Developing an IS as a New Business

Pedro Melo Vasconcelos

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Miguel Leitão Bignolas Mira da Silva
Prof. Luís Mira da Silva

Examination Committee

Chairperson: Prof. Nuno João Neves Mamede
Supervisor: Prof. Miguel Leitão Bignolas Mira da Silva
Member of the Committee: Prof. Paulo Jorge Fernandes Carreira

November 2015
Acknowledgments

This thesis marks the closing of an important period in my life during which I absorbed an amazing amount of knowledge both on an academic and personal levels. I feel I grew immensely over the last few years and am now a better human being and more capable of effectively contributing to society. Largely responsible for this are the people who accompanied me on this journey. From my parents and friends, to teachers and classmates, student groups I’ve been part of such AEIST and foreign students I’ve met in exchange programs like Athens. This is my thank you to all.

First, I would like to express my gratitude to my supervisor, Professor Miguel Mira da Silva for his tireless dedication to his students and the work the does at Instituto Superior Técnico - a proximity to students, future students, and the professional community that few are able to replicate. His dedication, support and guidance were crucial in successfully completing this work.

I would also like to express my gratitude to all the people who were involved in the project that was used as a demonstration for this thesis. A special thanks to Luís Mira da Silva, my co-supervisor, Filipa Sacadura, Teresa Vaz, Jorge Marques and all the remaining members of the team at Inovisa who welcomed and collaborated with us and made this project possible. Also involved in the project were Paulo Abreu and my colleague André Machado, who I would like to thank for the valuable help in developing and discussing parts of the project’s software.

Finally, a big and special thank you to my parents whose support and motivation where nothing short of phenomenal.

Thank you to all of you who made this journey possible.
Resumo

Sistemas de Informação (SI) são cada vez mais uma parte fundamental da nossa sociedade. Outrora utilizados pelas organizações como forma de optimizar operações e reduzir custos, representam agora parte integral de muitos produtos e serviços. Apesar disto, as taxas de insucesso no desenvolvimento destes sistemas mantém-se elevadas. Mais ainda, alguns autores reconhecem que a aplicação dos tradicionais critérios de sucesso de duração, custo e âmbito a este tipo de projectos, não é suficiente e que podemos acabar por desenvolver um sistema que não acrescenta valor para o seu público-alvo nem beneficia a organização.

Defendemos que o Valor de um SI deve ser um critério de sucesso adicional a ser usado nestes projectos. Propomos uma definição para Valor de um SI, juntamente com uma metodologia de desenvolvimento baseada em Scrum, que usa esta definição para guiar o desenvolvimento do SI no sentido de aumentar o valor do sistema. Na definição de Valor do SI recorremos ao trabalho de DeLone e McLean na área do Sucesso de SIs. Para guiar o desenvolvimento do sistema com base neste Valor, criamos uma camada sobreposta ao Scrum, baseada na metodologia Lean Startup para desenvolvimento de novos negócios e produtos.

A demonstração desta proposta consiste no desenvolvimento de um SI para partilha de conhecimento nos sectores da Agricultura, Alimentação e Floresta – a Plataforma SKAN. A avaliação global da proposta é positiva; mostramos que o Valor de um SI é uma dimensão útil do sucesso destes projectos e que pode ser usado para guiar o seu desenvolvimento.

Palavras-chave: Desenvolvimento de Sistemas de Informação (SI), Desenvolvimento Ágil, Lean Startup, Sucesso em projectos de SI, Valor de um SI
Abstract

Information Systems (IS) have become increasingly pervasive in our society. Organizations employ them not only to streamline operations but also as a core component of their products and services offerings. Despite this, failure rates for IS projects have remained consistently high over the years and moreover, some authors recognize that applying just the traditional success criteria of time, cost and scope to these projects, may well lead to a system that is not useful for its intended users nor benefits the organization.

We argue that IS Value should be used as an additional dimension of project success. We propose a definition of IS Value and a development methodology, based on Scrum, that uses this definition to guide the development efforts towards increasing the value of the system. For an IS Value definition, we draw on the work of DeLone and McLean on IS Success. To guide development according to IS Value, we create a new layer on top of Scrum that is based on the concepts from the Lean Startup methodology for business and product development.

We provide a demonstration of this proposal in the form of the development of an IS for knowledge sharing in the Agriculture, Food and Forestry sectors – the SKAN Platform. The evaluation of this proposal is overall positive and we show that IS Value is a useful dimension of project success that can be used to guide development.

Keywords: Information Systems Development, Agile Software Development, Lean Startup, IS Project Success, IS Value
Contents

Acknowledgments ................................................................. iii
Resumo ................................................................. v
Abstract ................................................................. vii
Contents ................................................................. x
List of Tables ................................................................. xi
List of Figures ................................................................. xv
Glossary ................................................................. xvii

1 Introduction ................................................................. 1

2 Research Methodology ................................................................. 3
2.1 Motivation for using DSR Methodology ................................................................. 3
2.2 An Overview of the DSR Methodology ................................................................. 4

3 Problem ................................................................. 7
3.1 Overview ................................................................. 7
3.2 Motivation ................................................................. 8
3.3 The Problem in Detail ................................................................. 9
   3.3.1 The high failure rates for IS development projects ................................................................. 9
   3.3.2 The inadequacy of IS project success definition ................................................................. 10
   3.3.3 Research problem statement ................................................................. 14

4 Related Work ................................................................. 15
4.1 The Plan-Driven approach ................................................................. 15
   4.1.1 Early Information Systems – The Rise And Fall Of SDLC ................................................................. 16
   4.1.2 Value Based Software Engineering ................................................................. 17
4.2 The Agile approach ................................................................. 18

5 Objectives ................................................................. 21

6 Proposal ................................................................. 23
6.1 Theoretical Background ................................................................. 23
   6.1.1 The Scrum Methodology ................................................................. 23
List of Tables

3.1 Plan execution versus solution discovery: bridge construction versus online social network development ................................................................. 13

7.1 Technical approaches considered for SKAN Platform implementation ........................................ 37
7.2 Task 1 details - Profiles and search (Users) ................................................................. 45
7.3 Task 2 details - Upvotes and downvotes (Posts/Comments) .............................................. 46
7.4 Task 3 details - Messaging between users MVP (Messaging) ............................................. 47
7.5 Task 4 details - Content creation (Posts) ........................................................................ 48
7.6 Task 5 details - Calendar view (Events) ........................................................................ 49
7.7 Task 6 details - Messaging between users (Messaging) ..................................................... 49
7.8 Task 7 details - Notify users of important events (Notifications) ...................................... 50
7.9 Task 8 details - Weekly newsletter (Newsletter) ............................................................... 51
List of Figures

2.1 The DSRM Process Model [1] ......................................................... 4
3.1 IT capital investment in the US as percentage of total invested capital [2] ......................... 9
3.2 Success rates of IT projects (data from 2012) [3] ............................................. 10
3.3 The Iron Triangle representing traditional dimensions of project management success .... 10
3.4 Plan Execution versus Solution Discovery .................................................... 12
6.1 The Build-Measure-Learn feedback loop [4] .................................................. 26
6.2 The Delone & McLean IS Success Model [5] .................................................... 27
6.3 A new approach to IS project success ............................................................. 29
6.4 Solution proposal - Product release and customer feedback frequencies .......................... 30
6.5 The Scrum and Lean Startup methodologies combined ...................................... 31
6.6 The Scrum and Lean Startup methodologies combined ...................................... 32
6.7 Each Sprint produces Data to be used as input in subsequent Sprints ....................... 33
7.1 Simplified Organization Breakdown Structure (OBS) for the SKAN Platform project .... 36
7.2 The development of country networks spread over three stages in the project roadmap .... 38
7.3 Identical standalone networks are aggregated in SKAN Platform ............................ 39
7.4 SKAN Platform project timeline .................................................................... 40
7.5 Team Structure ......................................................................................... 41
7.6 Active users between Oct 1, 2014 and Aug 31, 2015 ........................................... 42
7.7 Number of webpages visited between Oct 1, 2014 and Aug 31, 2015 ..................... 43
7.8 Average time spent on a given webpage between Oct 1, 2014 and Aug 31, 2015 .......... 43
7.9 Users who registered for an account between Oct 1, 2014 and Aug 31, 2015 ............... 44
7.10 Cumulative and total registered users between Oct 1, 2014 and Aug 31, 2015 ............ 44
7.11 Networking among users as measured by user lookups, profile views and exploring the users index between Oct 1, 2014 and Aug 31, 2015 ............................................... 46
7.12 Upvotes and downvotes on posts between Oct 1, 2014 and Aug 31, 2015 .................. 47
7.13 Upvotes and downvotes on comments between Oct 1, 2014 and Aug 31, 2015 ........... 47
7.14 Posts by development team members, total number of posts (including by users) and average time on page between Oct 1, 2014 and Aug 31, 2015 (Avg. Time on Page plotted on secondary axis on the right) ......................................................... 48
A.13 Notifications dashboard. Users are notified of new messages, new comments on their posts and new upvotes or downvotes on posts and comments. In addition to these notifications, users can also be notified via email, according to preferences they can change in their user settings.

A.14 Administrator settings for a given SKAN network. Administrators can edit network-specific settings such as network name, logo and simplified html content.

A.15 Cookie policy. When visiting the website for the first time, new users are informed about the usage of cookies and the website’s cookie policy.

A.16 Administration interface. SKAN administrators have access to a special interface where they can perform multiple administrative tasks such as delete bad posts or fake users.
Glossary

DSRM  Design Science Research Methodology
IS    Information System
LSM   Lean Startup Methodology
MVP   Minimum Viable Product
SDLC  Systems Development Life Cycle
VBSE  Value Based Software Engineering
VP    Value Proposition
Chapter 1

Introduction

Information Systems have come to play an increasingly important role in every aspect of our personal and professional lives. The idea of Information Technology as an isolated tool, used for the sole purpose of providing operational support in organizations, is fading away. More and more, organizations now see Information Systems as a core aspect of their businesses, essential not only in supporting operations, but also as part of their products and services offerings [6].

Despite this, ISs failure rates remain consistently high and many of these systems end up not being used or otherwise not delivering on expected value [3, 7, 8, 9].

This research aims at better understanding the problem of ISs projects failure and at providing a solution to increase ISs success and value delivered to customers.

In Chapter 2 we present the methodology used in this research - Design Science Research Methodology - and the reasons for this choice.

In Chapter 3 we discuss the problem under consideration. On a superficial level we identify the problem as consisting in the high failure rates for IS development projects. A closer inspection, however, reveals the definition of success for IS projects to be a tricky concept. Most of the time this definition defaults to the standard used in project management in general: the iron triangle of cost, time and scope. On the other hand, some authors recognize that a project can still meet the goals defined for these three criteria while at the same time not being used by customers, not liked by sponsors and not beneficial for the organization. In the second half of this chapter we explore the kind of challenges IS projects tackle and conclude that many of them are projects where requirements are very uncertain and exploring the solution space is or should be the main activity. We end this chapter by extracting a problem statement from the previous analysis. In summary we can say that, by relying on traditional project success criteria, the prevalent IS development methodologies do not help in guiding development efforts towards increasing the value created by the system.
Chapter 4 describes existing work related to the problem presented in 3. Approaches to the problem of how to best develop Information Systems are usually described as development methodologies or practices. We cover two major approaches: the plan-driven approach and the agile approach. Inside the plan-driven category we explore a proposal that explicitly deals with the problem of guiding development efforts based on the value of the system being developed.

In Chapter 5 we put forward the objectives of a potential solution to this problem. These objectives are based on the problem definition presented in Chapter 3 and in the related work presented in Chapter 4. We argue that a solution should provide a definition of IS Value that could be incorporated in the success criteria for IS development projects, and that it should also provide a methodology that uses the new project success criteria to guide the development.

Chapter 6 is where we present our proposal. We dedicate the first part of this chapter to the theoretical background used as a basis for the proposal. The second part is where the proposal is described. In a simplified way, the proposal can be described as an IS development methodology, based on Scrum, that uses a measure of IS Value to guide the development towards increasing the value delivered to customers. We use DeLone and McLean’s work on IS Success to define IS Value and incorporate this dimension in the traditional view of project success. We propose a mechanism that can be layered on top of Scrum to continuously measure both system-wide and feature-specific value and use this information to guide the development of the Information System.

Chapter 7 corresponds to the demonstration of our proposal. This consisted of a case study where we implemented the proposal in a real world scenario - the development of the SKAN Platform IS. This was a project backed by multiple stakeholders such as ISA, UÉvora, UTAD, IICT and INIAV, with the goal of developing a platform for knowledge sharing in the Agriculture, Food and Forestry sectors. We start the chapter by providing context on the SKAN initiative, stakeholders, goals and initial discussions. The second part of the chapter is dedicated to the demonstration of the proposal. Here we detail three core parts of its implementation: the definition of core value propositions for the project, the definition of the project-specific aspects that characterize IS Value and finally, the development work, which takes place in development cycles or Sprints.

In Chapter 8 we evaluate our proposal. This evaluation, based on methods proposed by different authors for evaluation of Design Science, is carried out in three ways: An analysis of our empirical observations during the case study demonstration, interviews with participants in the case study and finally, an analysis of design-oriented research principles.

In Chapter 9 we end with a conclusion where main contributions, limitations and future work are discussed. By providing an alternative way to look at the success of IS projects together with a measure of IS Value and a methodology to guide development towards increasing this value, we think our proposal addresses the problem identified in this research and can contribute to improve many of today's IS development projects.
Chapter 2

Research Methodology

In this research we have used the Design Science Research Methodology (DSRM). This section provides an overview of the methodology and motivation for such choice. We end this section with a simple diagram mapping the generic steps in the methodology with the concepts specific to this research.

2.1 Motivation for using DSR Methodology

DSRM gained popularity as a reaction to a problem identified in the Information Systems (IS) research discipline. The problem was that IS research is supposed to be an applied research discipline where theory from other domains of knowledge such as economics, computer science or social sciences, is applied to solve problems involving Information Technology and organizations. In other words, the research must be applicable to the real world and solve real problems. Instead, however, dominant paradigms in IS research were predominantly explanatory and descriptive, influenced by other more traditional disciplines such as the social and natural sciences [1].

As Peffers et al. argue, by not producing explicitly applicable research solutions, the IS research discipline risks losing relevance and influence over neighboring research streams such as requirements engineering or software engineering. Peffers et al. explain that while design, as the act of creating an explicitly applicable solution to a problem, is an acceptable approach in research disciplines such as engineering, it has not gained traction in IS research [1].

For some researchers though, studying and understanding the way things are is not enough; they want to understand how they can improve them. It is in this context that DSRM is born as a new paradigm in IS research, oriented towards problem-solving and well suited to address the so called wicked problems [10].

According to Peffers et al., Design Science creates and evaluates IT artifacts intended to solve identified
organizational problems [1], which is one of the main goals of this research proposal.

Wicked Problems Another motivation for using the DSRM is the fact that this research is centered around what some would call a wicked problem (described in Section 3 – Research Problem).

As Hevner et al. explains, natural science research methods are appropriate for the study of existing phenomena but they are insufficient for the study of what he calls “wicked organizational problems, the type of problems that require creative, novel and innovative solutions”. These problems are characterized by unstable requirements, conflicting views among stakeholders, ill-defined environmental contexts and complex interactions among sub-components of the problem.

Design Science positions itself as the paradigm shift required to effectively address such problems.

2.2 An Overview of the DSR Methodology

The DSRM is an iterative process composed of six activities, which we will briefly explain here (summarized in Figure 1). In order to understand the motivation for each activity, one must keep in mind the goals of the methodology. As previously mentioned in this document, Design Science can be summarized as follows:

“Design Science creates and evaluates IT artifacts intended to solve identified organizational problems.”

![Figure 2.1: The DSRM Process Model [1]](image)

**Activity 1: Problem identification and motivation** The first activity in the DSRM consists of identifying the organizational problem for which a solution is to be designed. In this activity, the problem must be clearly identified and conceptually sub-divided so that a solution can capture its complexity. It is important to explain the value of a solution in order to provide motivation for the audience. Requirements for this activity include: knowledge of the state of the problem and the importance of its solution.
**Activity 2: Definition of the objectives of the solution**  In this activity we infer the objectives of a solution based on the problem definition and in current solutions. Objectives can be quantitative or qualitative, such as a description of how a new artifact is expected to support solutions to problems not addressed before by existing solutions. Requirements for this activity include: knowledge of the state of the problem and current solutions.

**Activity 3: Design and development**  In this activity we create a solution. More specifically, an artifact must be produced. According to Peffers et al., “a design research artifact can be any designed object in which a design contribution is embedded in the design”. Artifacts are potentially constructs, models, methods or instantiations. This activity includes both determining the artifact’s desired functionality and creating the artifact. Requirements for this activity include: knowledge of theory that can be incorporated in a solution.

**Activity 4: Demonstration**  Here we must use the artifact created in the previous activity and apply it to solve one or more instances of the problem. This may include its use in simulation, case study, proof, or other appropriate activity. Requirements for this activity include: knowledge of how to use the artifact to solve the problem.

**Activity 5: Evaluation**  In this activity we must observe and measure how well the artifact contributes to solving the problem. The objectives of a solution put forward in Activity 2 must be compared with observed results of using the artifact in the demonstration (Activity 4). Depending on the nature of the problem and the artifact, evaluation can take different forms such as satisfaction surveys, customer feedback, assessments of budgets and items produced, among others. The goal is to provide appropriate empirical evidence or logical proof and to decide whether or not to iterate back to Activity 3 to try to improve the effectiveness of the artifact.

**Activity 6: Communication**  In this last activity, both problem and solution must be communicated to the community, thus fulfilling one of the major DSRM goals of advancing the IS field by contributing with applied research. One must communicate the problem and its importance, the artifact, its utility and novelty, its design and effectiveness.
Chapter 3

Problem

In this section we describe the first step in the DSRM, which entails identifying the problem under consideration and providing motivation for its study.

3.1 Overview

The most easily recognizable part of the problem lies with the high failure rates for IS Development projects. Despite increased demand for Information Systems, there is a consensus in the literature over the fact that failure rates for this kind of projects have remained consistently high over the years. A significant portion of these newly created systems ends up not being used at all or otherwise failing to deliver on expected value [3, 8, 9].

The complexity of the problem is further increased by the difficulty in measuring the success of IS projects. What is success? For whom? Is it enough for a project to be completed under time, budget and scope constraints for it to be considered successful? Even if nobody uses the resulting system?

Success has multiple dimensions with different weights for each stakeholder. We can refer to project management success, technical success or business success [11]. IS success is a very important concept and yet a poorly defined one in many projects. It is possible, for instance, for a development team to consider a project successful while at the same time, from a business perspective, the resulting system to be found of little or no value.

The issue is extremely relevant because Information Systems have become increasingly important in our society. More and more, organizations now see Information Systems as a core aspect of their businesses, essential not only in supporting operations, but also as part of their products and services offerings [6]. It is thus, important, to find ways that allow us to maximize the value derived from developing such systems.
3.2 Motivation

**A Definition of Information Systems** It is important that we start by clarifying what we mean by Information System (abbreviated as IS). In this document, when talking about Information Systems in general, we abide by the broad definition presented in Encyclopedia Britannica (2008), which describes an IS as “an integrated set of components for collecting, storing, processing, and communicating information. Business firms, other organizations, and individuals in contemporary society rely on information systems to manage their operations, compete in the marketplace, supply services, and augment personal lives.” (Encyclopedia Britannica, 2008).

Although not explicit in the paragraph above, the definition of IS we use here has, among others, two very important components: An Information Technology (IT) component and a people component. As we will later argue, understanding ISs as more than their IT components is essential when talking about their failure or success.

Examples of the ISs we are referring to include systems ranging from the more enterprise-oriented such as Enterprise Resource Planning (ERPs) and Customer Relationship Management (CRM) systems, to e-commerce systems, to the more hedonistic (used for recreation) types of systems, such as social networks or audio and video streaming services.

**The impact of Information Systems in organizations and in our lives** Having described what we mean by ISs, we must now explain why failures to create or modify such systems have a tremendous impact in organizations and in our lives in general – and the reason lies in the increasing demand and pervasiveness of these systems across almost every industry.

The first ISs, as described in the paragraphs above, emerged during the 1950s along the introduction of the first computers as tools for automation and optimization of tasks such as accounting and record-keeping [12]. During the 1960s, these systems evolved in complexity, giving rise to “(...) large-scale functional business systems in an age of large-scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines.” [13].

Information Systems continued to evolve both in complexity and areas of application. By the 1980s, IT capital investment accounted for 32% of all capital invested. By 2009, the amount of capital invested solely in IT had reached 52% of the total invested [2].

This trend for IT spending seems to continue as more and more organizations start viewing ISs as a core aspect of their businesses, essential not only in supporting operations, but also as part of their products and services offerings [6]. A Gartner Report from 2013 identifies “expanding technology” as one of the top 5 dominant global business strategies [6]. Looking at the top 10 global technology priorities of CIOs, the same study indicates a focus on customer-oriented technology as opposed to traditional operations-oriented ISs [6].
Some of the reasons for these trends have been explained before in the literature and seem to continue to be valid in the foreseeable future: Organizations operate in an increasing globalized and competitive environment where agility in reacting and seizing opportunities becomes an imperative for success. ISs provide significant opportunity for enhancing agility [14].

Such trends help explain both the increased demand for ISs in organizations and their growing pervasiveness and impact in our lives – while playing a crucial role in supporting the operations of organizations, ISs’ applications expand even further as they become embedded in organizations’ products and services offerings. Regular customers become direct users of these systems or, as we argue, more than users, they become part of these systems.

Motivation Summarized High failure rates in ISs projects represent a major problem because these systems have become increasingly pervasive with great impacts across a broad range of applications. Studies indicate that these systems represent over 50% of the total amount of capital invested in the US [2] and the demand seems to be rising world wide as “expanding technology” is seen as one of the top five dominant global business strategies for 2013 [6]. The success of ISs projects is important for organizations because the capital investments in these projects represent more than half of the total amount invested.

3.3 The Problem in Detail

3.3.1 The high failure rates for IS development projects

Although an exact figure is hard to provide, there is consensus across the literature over the high failure rates for ISs projects [3, 8, 9, 15], with estimates of project failure rates varying between 50% and 80% for large IS projects [16]. The widely cited Chaos Manifesto points to an overall ISs project success rate
of only 39% in 2013 [3].

In section 3.2 we have explained why the success of ISs projects is important for organizations. The most easily recognizable part of the problem we are tackling is then the fact that a large share of about 60% of ISs development projects is considered to be unsuccessful.

As implied before in this document, this is a widely recognized problem in the ISs literature. In Section 4, Related Work, we look into what solutions have already been proposed to solve it. In order to understand the context in which these solutions were proposed, however, we must first study our problem a little deeper.

Although the problem definition above points to an undesirable state of affairs in the ISs development field, it hides underlying assumptions over the meaning of project success.

### 3.3.2 The inadequacy of IS project success definition

The definition used in the widely cited Chaos Manifesto describes successful ISs projects as projects "delivered on time, on budget, with required features and functions" [3].

This definition is also in line with the definition used for projects in general in the Project Management Body Of Knowledge which states that “(…) the success of the project should be measured in terms of completing the project within the constraints of scope, time, cost, quality, resources, and risk as approved between the project managers and senior management.” [17].

![Figure 3.2: Success rates of IT projects (data from 2012) [3]](image)

![Figure 3.3: The Iron Triangle representing traditional dimensions of project management success](image)
The Project Management Triangle in Figure 3.3, also known as the Iron Triangle, represents the dimensions along which project success is traditionally measured – Scope (often including quality-related aspects), Time and Cost. The great majority of project management definitions describe the top goal of project management as completing the project within constraints defined around these three dimensions [18].

The problem with using these three dimensions as sole criteria for success is that they may not help in delivering a project that really benefits its stakeholders. As Atkinson notes, a project can be implemented on time, within budget and to agreed upon requirements, but still not be used by customers, not liked by the sponsors and not providing improved effectiveness or efficiency for the organization [18]. Many times, the potential value of ISs projects is not delivered and the success criteria used to evaluate them do not contribute to solve this problem.

In summary, according to the most commonly used criteria, a project can be considered a success in the short term because it was delivered according to the initially defined constraints on scope, time and cost, while still not providing any value for its stakeholders. But how can this happen?

Focus on the execution of a solution versus focus on the discovery of a solution

In short, ISs projects can be considered successful while not providing much value for stakeholders because the success criteria most commonly used to assess them do not measure value creation. They encourage a strong focus on ensuring conformity to plans but they provide little motivation to spend the necessary resources checking if initial plans really answer customer needs. We can view it as a focus on the execution of a solution versus a focus on the discovery of a solution. A plan can be successfully executed while still being the wrong plan. The problem is made even worse because, as we explain in the example below, ISs solutions seem to be increasingly less clear a priori while, at the same time, easier to execute.

What we argue is that by using scope, time and cost as the sole criteria for an ISs project success, there is little motivation to “correct the plan” from the project manager perspective. This responsibility is left to the customer and he may not be in the best position to undertake it.

Such mechanisms to “correct the plan” and somehow increase the value delivered by the project, could eventually be embedded in a ISs development methodology but there is no motivation for doing so due to the success criteria being used. At the same time, it is recognized that value-oriented shortfalls represent the major causes for ISs failures [19].

Although this may seem a little strange, it is not hard to see why this is not such a big problem in other fields of engineering. Let us consider an example in the next section.

Why building a bridge is different from building an online social network

In a case study from 1986, Spector et al. popularized the idea of comparing civil engineering with “computer systems design” by investigating in detail the process of bridge design [20]. The paper argued that both design processes
shared a lot of similarities and that the most notable difference was that the bridge design process was much more structured than the computer design process – it had more checks and inspections, more standards, greater design efforts and greater variety of tools. It is interesting to note that there is not a clear reference to the challenges and methodologies being radically different in both fields. While recognizing that advancements in technology would lead to a greater diversity than that which existed in the bridge design process, the authors suggest that as a mature engineering field, bridge design might provide a glimpse of the future of computer systems design.

One must keep in mind that the great majority of today’s most popular software development tools and programming languages did not exist at this time. Nothing similar to what we call the Internet, existed at this time. The technologies supporting ISs have changed dramatically but the methodologies for developing such systems have not.

**Plan-execution versus solution-discovery projects** Figure 3.4 depicts in a very simplified way the different challenges between building a bridge and developing an online social network. For the bridge we see that it is easier to predict the value in building it and the main challenges rely on the technical aspects of the project; this is a case where it makes sense to spend a large amount of time in the planning stage; we can say it is more of a *plan-execution* project. For the social network, however, the opposite is true; while the technical aspects can be said to me more straightforward, it is hard to predict its value; it is hard to say whether people will want use it or not and it is hard to select the features that will best address the needs of its potential users; in this case we say this is more of a *solution-discovery* project.

![Figure 3.4: Plan Execution versus Solution Discovery](image)

Here we find the other piece of the problem: using the Iron Triangle as the sole criteria for measuring ISs projects success shifts focus away from the creation of value for the customer. The focus is more on executing a plan and less on discovering a solution.

Table 3.1 helps us understand the nature of the problems we try to solve with ISs. Many of these problems are considered wicked problems, with conflicting views among stakeholders and ill-defined
Bridge Online Social Network

<table>
<thead>
<tr>
<th>Stakeholders</th>
<th>Sponsor, local citizens, tourists.</th>
<th>Besides the sponsor, characterizing potential users is a lot less straightforward than in the bridge example.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added value for stakeholders</td>
<td>The problem of grasping the value added by the bridge is essentially the problem of estimating the need for people to cross the bridge at the moment and some time in the future. Although it may not be a trivial problem, benefits are intuitively easier to grasp and estimates easier to produce. The risk of these estimates becoming invalid in a near future is low because alternatives to the bridge are not easily developed. While it is easier to define criteria for project completion (namely the scope), the value of the project is delivered at once (e.g.: the bridge is either complete or it is unusable).</td>
<td>The problem of grasping the value is usually a very hard one and not tractable. There are multiple stakeholders, conflicting requirements, and the risk of alternative solutions being quickly developed is very high. Defining a solution is harder but providing incremental value is easier (e.g.: the online social network is not yet complete with all planned features such as private messaging, but already provides value because you can lookup friends and create posts).</td>
</tr>
<tr>
<td>Difficulty in technical implementation</td>
<td>As was true 30 years ago, bridge design still requires a significant deal of customization for specific solutions. There are a lot of regulations to be followed and security concerns to attend to. The project is life critical and careful planning is required.</td>
<td>As is true of many IS projects, building an online social network is not a life-critical project. Today’s technologies allow for very fast development times and it is possible to create an early but functional version of the product in a matter of days.</td>
</tr>
</tbody>
</table>

Table 3.1: Plan execution versus solution discovery: bridge construction versus online social network development

requirements in fast changing environments.

Multiple authors recognize IS development as a wicked problem [21]. In [21], Nerur et al. recognize IS problems as being unique and difficult to formulate with solutions continually evolving as the designer gains a better appreciation of what must be solved. The fast pace at which IT evolves amplifies the problem even further: not only the solutions change as the designer’s understanding of the problem changes, but what is possible also changes continuously. Spending an extensive amount of time in planning may then become futile for many ISs projects because by the time we reach an implementation step, the problem definition has probably changed already and so have the available approaches to the problem.

We incur the risk of developing an inadequate system and we delay the delivery of value for much more than needed.
3.3.3 Research problem statement

In the previous two sections we explored two aspects of the problem: The first is that success rates for IS development projects are considerably low at around 39%. The second is that the success criteria applied to these projects do not measure the value created by these projects nor they help guide the development towards increasing this value.

Finally, by combining the different parts of the problem explored in the previous two sections, we can summarize our research problem as follows:

Research problem: By relying on traditional project success criteria, namely time, budget and scope, the prevalent IS development methodologies do not provide help in guiding the development efforts towards increasing the value of the system being developed.
Chapter 4

Related Work

In Section 3 we have identified the problem and provided motivation for its study, which corresponds to the first step in the DSRM. The second step of the DSRM is the definition of the objectives of a solution and will be dealt with in Section 5 – Objectives. The present section represents a bridge between the first and second steps of the DSRM where we will explore existing solutions for the problem. Understanding how these solutions do or do not solve different aspects of the problem will make it easier to define objectives for a new solution in Section 5.

The most common solutions to the problem of high failure rates of ISs projects come in the form of development methodologies or guiding principles for creating such systems. The main distinction we make between proposed solutions is in terms of more traditional approaches characterized by a heavy focus on planning (the Plan-driven approach) versus more flexible approaches such as the ones following Agile principles (the Agile approach).

4.1 The Plan-Driven approach

This is the type of solution characterized by a heavy focus on planning. This type of solution falls prey to the sub-problem identified in Section 3, which is trying to tweak success factors (such as methodologies) in order to help attain the success criteria, while considering the wrong (or incomplete) success criteria. This is the type in which structured approaches to ISs development, such as the SDLC family of methodologies, are included. Examples of such methodologies are the Waterfall, the V-Shaped, the Incremental Life Cycle or the Spiral models. These methodologies are characterized by having clearly defined and distinct work phases such as planning, designing, building, testing, and delivering a system.
4.1.1 Early Information Systems – The Rise And Fall Of SDLC

Many early Information Systems of the 1960s and 1970s were developed without explicit development methodologies. Emphasis was placed in the programming aspect of the development; developing an IS was mainly a software problem – most challenges were technical challenges [22].

Developers of these early systems were trained in computer technology and rarely understood the business context in which these systems were implemented. Both user and business needs were many times left unanswered. Projects were often poorly managed and poorly controlled. This state of affairs led to an appreciation of more disciplined approaches to ISs development and the first methodologies were born, hoping to improve the development process and increase the value these project could bring to organizations.

This early approach to ISs development came to be known as the Systems Development Life Cycle. SDLC is used to describe a family of methodologies of which the Waterfall Model is one of the most popular examples. SDLC methodologies are characterized by following a sequence of clearly defined activities such as feasibility study, analysis, design, development, implementation and maintenance; each activity can start only after the previous had been completed [22]. The key concept in SDLC is that there is a well-defined process by which an IS is conceived, developed and implemented [23].

The Waterfall Model is probably not going away anytime soon [24] and, in fact, plan-driven approaches may very well be suited for certain kinds of projects of which life-critical systems are an example [25]. Nevertheless, the limitations of plan driven approaches as the standard way to develop ISs have long been identified. Fitzgerald [22] lists some of the most pressing issues:

- Failure to meet the real needs of the business;
- Overly conservative systems design;
- Lack of adaptability to changing businesses and markets;
- Inflexibility, making changes in the design difficult and costly;
- User dissatisfaction due to inability to test the system before it is operational;
- Outdated and not very user-friendly documentation;
- Heavy workload caused by attempts to change the system to reflect changing user needs;

Fitzgerald also identifies some key assumptions methodologies may wrongly take:

- That the environment is stable;
- That users are knowledgeable about their own requirements;
• That consensus over requirements can be achieved;

Ultimately, it is not the case that plan-driven methodologies are necessarily wrong. Rather, the development community agreed that there were better alternatives. There were ways to increase value creation and to do it faster, for a broad set of problems. The type of problems we are talking about are the wicked problems we’ve defined in Section 2. These are problems for which heavy planning is not feasible because they often have conflicting, ill-defined, and extremely volatile requirements.

The SDLC and plan-driven approaches in general were recognized as inadequate to tackle these problems. More flexible approaches were favored, which embraced change and focused on quicker delivery of value to the customer [26]. The Agile movement emerged from the insight that “the opportunity for competitive advantage comes from being more agile than the competitors in one’s market” [25].

Plan-driven methodologies are not adequate to solve the problem under consideration because they are not oriented towards the creation of value for the customer. They operate under the standard success criteria of time, budget and scope. They wrongly assume that achieving a plan implies project success and value creation [26] and in the last years, for many projects, they have been dismissed by practitioners in favor of more flexible approaches.

4.1.2 Value Based Software Engineering

While Section 4.1.1 represents the most popular solutions in the plan-driven approach to IS development, there are alternative solutions. One of these alternative solutions recognizes a key problem with most ISs development methodologies. The problem is that the development process is value-neutral – every requirement, use case or defect, is treated as equally important [19].

This approach is called Value Based Software Engineering (VBSE) and it addresses the same problem we have identified in this research: ISs projects failure rates remain consistently high and most failures are caused by value-oriented shortfalls.

VBSE argues that value considerations should be integrated in the IS development process which is precisely the idea behind the solution we propose in Section 5. The differences between VBSE and the solution we propose lie in how value is integrated in the development process – VBSE proposes to accomplish this integration through 7 key elements:

• Benefits Realization Analysis;

• Stakeholder Value Proposition Elicitation and Reconciliation;

• Business Case Analysis;

• Continuous Risk and Opportunity Management;
• Concurrent System and Software Engineering;
• Value-Based Monitoring and Control;
• Change as Opportunity.

Although we will not go into the detail of each of the elements, they were presented here to provide a sense for what this approach entails. Its goal is to inform the development process with value creation aspects but its methods remind us of other plan-driven approaches.

The literature on VBSE is scarce and the approach does not seem to have gained any traction. We speculate that this is in part due to its overheads reminiscent of other plan-driven approaches – some kind of plan (such as benefits and risk analysis) is still being created and some metrics (such as Earned Value) are still being used in a way that does not clearly represent added value for the customer.

VBSE is interesting because it recognizes the benefits of guiding the IS development by using value considerations. One such example is the prioritization of features based on the benefits they are expected to provide. In this example, the problem lies in the techniques used to estimate and measure this value. Together with other aspects of this approach, they seem to make the methodology more complex than it needs to be and we find ourselves in a position similar to where we were with SDLC approaches in section 4.1.2. – effort is wasted in unnecessary planning and value delivery is delayed.

4.2 The Agile approach

This type of solution departs from the mindset dominating the Plan-driven approaches. This type of solution emerges precisely as a reaction to that kind of approach. It does not negate the usefulness of some degree of planning and documentation, for instance, but it recognizes the limitations of these activities in solving the problems at hand [26]. This type of solution usually follows the principles laid out in the Agile Manifesto. Priority is given to customer satisfaction and early and continuous delivery of value.

Agile approaches to ISs development emerged in the early 1990s as a reaction to the inadequacy of plan-driven methodologies to tackle ISs problems [25]. Cockburn et al. use the analogy of a battlefield commander to explain the nature of the problems that IS were trying to tackle and the unfitness of plan-driven methodologies:

“No battlefield commander would say, “If we just plan this battle long enough and hard enough, and put repeatable processes in place, we can eliminate change early in the battle and not have to deal with it later on”” [25].
The type of problems for which Agile is suited Agile practices are applicable to a broad set of problems but specially when they have some of the following characteristics [25]:

- They use bleeding edge technology;
- They have conflicting and ill-defined requirements;
- Their requirements change as customers and development teams explore the unknown.

These are the characteristics of wicked problems and in Section 3 (Research Problem) we have argued that many of today's ISs problems fall into this category. This seems to make Agile approaches a good candidate for tackling the problem under consideration. In order to increase the value of ISs projects, they rely on a few tenets [26]:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation;
- Responding to change over following a plan;

These tenets were explained in detail in The Agile Manifesto, a manifesto published in 2001 by prominent professionals in the software and ISs development disciplines.

Just as VBSE (the solution presented in section 4.1.2), Agile approaches correctly identify the need to focus on value creation while developing solutions for these wicked problems. Agile approaches, however, address this need in a different way, advocating less management, less planning, and more customer interaction – a striking contrast with plan-driven methodologies.

Agile methodologies focus on value delivery to the customer but they provide no guidance as to the meaning of value. For example, in one Agile methodology, Scrum, this value is decided by the role of Product Owner, who is responsible for prioritizing development efforts. But the methodology lacks mechanisms for assessing the value of the system being developed and for guiding development efforts towards increasing this value - such decisions end up relying solely on the Product Owner's autonomous work or intuition. This can lead to a system agreed upon by a sponsor (or a Product Owner) but unwanted by users. The same happens for most other Agile methodologies besides Scrum.

In summary, Agile approaches are considered an improvement over Plan-driven approaches because of the flexibility they provide - it becomes easier to make experiments, to explore the solution space and to change requirements during development in response to environment changes. Agile approaches, however, do not completely address our problem because they are not explicit in terms of success criteria, they do not define IS value and do not provide mechanisms to guide development efforts towards increasing it.
Here is where there is room for improvement in the Agile approach  The solution we suggest in Section 5 builds on this idea of finding out how we can increase the value delivered by an IS.
Chapter 5

Objectives

This section corresponds to the second step in the DSRM – Definition of the objectives of a solution. In the previous section (Section 4 - Related Work), we have explored different existent solutions and the reasons why these solutions do not completely address the problem.

In this section we start by discussing the objectives of a solution based on the problem definition in Section 3. After this, we look into the solutions explored in the previous section to understand what lessons can be learned from existent solutions and how these lessons can contribute to define the objectives of a new solution.

From the problem definition in Section 3:

- The solution must provide a definition for IS value that can be incorporated in the success criteria for IS development projects.

- The solution must provide an IS development methodology that uses said success criteria to guide the development towards increasing the value of the system being developed.

From the lessons learned from existing solutions explored in section 4:

- The solution must guide the ISs development process while adhering to Agile principles – change must be embraced and value must be delivered to the customer sooner rather than later; Agile principles and methodologies provide a good base on which a solution can be built upon.

Given these objectives, we now proceed to Chapter 6, where we will explore useful theoretical background and propose a solution that fits our requirements.
Chapter 6

Proposal

This section corresponds to the third step in the DSRM – Design and Development. In the previous section (Section 5 – Objectives) we have outlined the objectives of a solution to the problem under consideration and we have also provided a basis from where to start designing a new solution.

In this section we start by presenting the theoretical background that shapes the solution we later propose. We end this chapter with a proposal for a solution.

6.1 Theoretical Background

6.1.1 The Scrum Methodology

Scrum is one of today’s most popular Agile methodologies. It emerged in the mid-1990s, under the assumption that ISs development is an unpredictable and complicated process that can only be described as an overall progression [27].

Scrum is based on the following concepts ¹ [28]:

**The Scrum Team**  The Scrum Team consists of a Product Owner, Development Team and a Scrum Master. The Scrum Team delivers products iteratively and incrementally – a potential useful version of the product is permanently available.

- **The Product Owner.** The Product Owner is responsible for maximizing the value of the product. He is also the sole responsible for the Product Backlog – this includes defining and prioritizing the work to be done.

¹We assume the reader to be familiar with the Scrum methodology and do not provide an exhaustive description here; emphasis is given to the concepts most relevant for the proposal presented in Section 6.2.
• **The Development Team.** The Development Team consists of professionals responsible for doing the work of delivering a potentially releasable increment of the product at the end of a Sprint.

• **The Scrum Master.** The Scrum Master is responsible for ensuring that the Scrum methodology rules and best practices are being followed. He works as both a leader and a servant to the whole team and his main goal is to facilitate the development process.

**Scrum Events**  Scrum events are used to create regularity and minimize the need and duration of meetings while improving communication and transparency.

• **The Sprint.** The Sprint is the central concept of Scrum. It is a time-box of varying length (usually between 15 days and one month) in which a potentially releasable product increment is created. Each sprint starts immediately after the conclusion of the previous sprint and consists of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective. The concepts of Sprint Planning and Sprint Review will be especially relevant in the context of our proposal for a solution in section 6.2.

• **Sprint Planning.** At the beginning of each sprint, the work to be done in that sprint is planned in the Sprint Planning meeting where the Product Owner chooses the items to be developed in the sprint, with input from the development team as to what is or is not feasible in the duration of the sprint. It is important to note that Scrum itself does not provide guidance on how the Product Owner should prioritize the backlog and on which tasks he should choose for development in the coming sprint. These decisions are seen as business decisions on which Scrum does not claim to provide help.

• **Sprint Review.** At the end of each sprint, the increment produced in the sprint is presented and discussed in the Sprint Review meeting. This meeting is also used to discuss changes to the backlog and possible next steps. Reflecting on the work from the previous sprint, the team can use this meeting to generate new ideas and make changes to the backlog. This meeting provides important input for the next Sprint Planning meeting.

**Scrum Artifacts**  Artifacts in Scrum are designed to provide transparency and maximize the understanding of key information.

• **Product Backlog.** The Product Backlog is an ordered list of every possible system requirement; this list is never complete – it is constantly changing in order to reflect what the Product Owner believes the product needs. He is responsible for the content and ordering of the Product Backlog.

• **Sprint Backlog.** The Sprint Backlog is a set of items selected from the Product Backlog to be developed during the sprint.
6.1.2 The Lean Startup

The Lean Startup [4] is a methodology proposed in 2011 by Eric Ries and it has been gaining considerable popularity as a means of developing new products and new businesses [29].

Similarly to what happened in the IS development field, traditional methodologies for business development rely heavily on business forecasts and plans. They assume it is possible to figure out the business unknowns in advance and too often, after months of development, entrepreneurs realize that customers do not need or want most of the product's features.

One of the key insights in the Lean Startup is that existing companies execute a business model while startups look for one. Existing companies execute solutions whereas startups are a solution discovery endeavor.

In section 3 we described IS problems as wicked problems and argued that a methodology for developing ISs cannot focus on the execution of a plan because the problem is not an execution problem but a solution-discovery one. The Lean Startup identifies the very same problem in the business domain – new companies spending a lot of effort in crafting their detailed business plans and executing them well, only to discover they were not the right plans - and proposes its own methodology for business and product development.

Inspired by lean manufacturing principles, the Lean Startup aims at minimizing wasted effort in the business development process. It relies on empirical evidence for measuring the success of the endeavor – progress is measured by what the organization learns about the shape of the solution it is building. These lessons are closely linked to the needs of the customers because ultimately it is they who decide the success of the product or business.

Key concepts in the Lean Startup methodology (LSM)

- **Build-Measure-Learn Feedback Loop and Validated Learning.** The Build-Measure-Learn is one of the most important concepts in the LSM. Instead of relying on assumptions, LSM uses evidence to learn and inform the development of the business.

  The Build-Measure-Learn feedback loop is what creates “validated learning” which is the unit of progress for startups. Validated learning is learning backed by empirical data collected from real customers. Since a business’ success depends on it having customers, Ries argues that what you learn about your customers’ needs and the ways to address them is what constitutes the success of a startup. It is what constitutes the success of the project of developing a new business. The project’s success increases as more information about the shape of the solution is discovered and validated.

  Hypothesis are defined that make predictions about what is supposed to happen. The most impor-
Hypothesis aim to test whether a certain feature really delivers value to customers using the product.

Hypothesis are tested with experiments. A Minimum Viable Product (MVP) is one such experiment – it is the most simple experiment one can think of that will provide evidence for supporting the hypothesis under consideration.

- **Lean Thinking.** Lean Thinking defines value as providing benefit to the customer. Benefit to the customer is measured in multiple ways in the Build-Measure-Learn feedback loop, including measures of usage and intentions to use the products. Here we are interested in understanding if we really are solving the customer’s problems or needs. The way he reacts to the solutions we provide is essential to understand if they are adequate or not.

The success in developing a business can be seen as the success in learning how to provide benefits to the customer because that is what will constitute the driver for future transactions between business and customer where value is delivered to both. We can say the development of the business or product is guided by the value it provides to the customer.

The fact that this process is carried out in an iterative and incremental way means that development cycles are shortened - the business can adapt faster and value can be delivered to the customer earlier in the process. It is up to the sponsors of these endeavors to decide when to switch from solution discovery to operations mode.

### 6.1.3 A measure of value – IS Success as defined by DeLone and McLean

First proposed in 1992, the DeLone and McLean IS Success Model [30] stands as one of the most popular frameworks for measuring IS success [5]. The more recent update of the model includes 6 dimensions along which IS success is measured.
Two of the key measures of success in the D&M IS Success Model are system usage and intention to use. The authors argue that even when the system usage is mandatory, variability in the quality and intensity of this use is likely to be a good indicator of the IS success. They also note that the relevance of system usage as a success measure is naturally increased if customer use is voluntary. They recognize that no single choice of success variables is intrinsically better than another – the organizational context and other factors must be taken into account.

6.2 Proposal For A Solution

We start this section by providing a summary description of the proposal and proceed to explain how the theoretical background explored before fits in the final proposal. In Section 6.2.1 we present the main characteristics of the proposal. In the remaining three sections, 6.2.2, 6.2.3, 6.2.4, we describe three sequential steps in which the proposal is implemented.

Briefly stated, the solution we propose can be described in the following way:

**Solution proposal:** An ISs development methodology that combines the Scrum methodology with the Lean Startup methodology and employs system use as measure of success for guiding the development process, thus increasing the value delivered by the system to its customers.

This proposal is built on top of the concepts explained in section 6.1 - Theoretical Background. Here we explain how they fit in the final solution proposal:

- **Scrum Methodology.** We recognize the wide adoption and usefulness of Scrum but we argue that leaving the creation and prioritization of tasks to the Product Owner alone, without providing additional guidance, can lead to a focus on the wrong set of features. The features the Product Owner feels are important are not necessarily the ones that customers value and want. Scrum brings the required agility to the development process but it does not provide guidance in terms of working towards increasing the business value of the development effort. We will use Scrum as the base structure for the development process, with an emphasis on Scrum’s events and artifacts.
A note on Scrum roles. In our solution proposal we use the same roles proposed in Scrum for the Scrum Team - Product Owner, Scrum Master and Team Member (Development Team). We want to clarify, however, our definition of Team Member and Development Team. In the original Scrum paper published by Schwaber in 1997 [27], he describes the Development Team as being composed of "developers, documenters and quality control staff" and it is not clear what he means by "developers". In our proposal we abide by the broader definition of "developer" provided by Sutherland [28] which describes the Development Team as being "cross-functional, with all of the skills as a team necessary to create a product Increment", composed of team members who are called Developers, "regardless of the work being performed by the person". We want to emphasize this because it is an important point in the shift from viewing IS Development as just the development of a working software product. Business and customer development, for instance, can also be useful skills to include in a Development Team.

• Lean Startup Methodology. We recognize the usefulness in the Lean Startup Methodology as a major shift in the way people develop businesses and products. The LSM focuses on finding a match between the products to be sold and the targeted market as the basis for a healthy business. Focusing on the end customer of the product is essential; its feedback is crucial in the process of developing the business or product. The process of gathering and using this feedback is captured in LSM's concept of the Build-Measure-Learn feedback loop. We will map this cycle into Scrum events with the intent of creating a version of Scrum that not only deals with project execution but also helps in guiding the development process by shaping the solution in a way that increases business value. In practice this means building a product that better fits customer needs.

• D&M IS Success Model. To evaluate the overall success of the Information System, we rely on DeLone and McLean's Information Systems Success Model, which suggests System Use as the main predictor of the information system’s net benefits or, in other words, the information system’s success.

We employ System Use as a proxy for measuring IS Value. This is applied both to the system as a whole and to specific features being developed.

With respect to the specific metrics used to evaluate system use, these can be varied, as suggested by Delone and McLean. Provided examples include "Number of site visits", "Number of transactions executed", "Amount of use", "Duration of use", "Number of inquiries", "Frequency of access", among many others [30] [5].

6.2.1 Main characteristics of the proposal

This proposal consists of four major aspects:

• It represents an IS development methodology based on Scrum and builds on top of it. The Scrum
Team, Events and Artifacts are kept and the development cycle is similar to the Scrum Sprint;

- It employs D&M's system *use* as a measure of value (or *key benefits*) of the system being developed;

- It incorporates value, as defined above, as another dimension of an IS project’s success; this means we now have an additional criterion to assess the success of an IS project besides time, budget and scope. Figure 6.3 depicts this new approach to IS project success as an evolution and addition to the traditional success criteria. IS Value must be defined explicitly in the form of aspects of the system that translate into system use (e.g. the time spent using the system or the completion of specific actions inside the system);

![Figure 6.3: A new approach to IS project success.](image)

- It adapts ideas from the Lean Startup methodology to build a new layer on top of Scrum. This new layer provides a mechanism to measure system value during development and use this information to guide the development efforts towards increasing the system’s value and also, consequently, its success.

On one hand, we consider Scrum as an improvement over plan-driven methodologies for the agility it provides. On the other hand, we see great value in bringing the customer closer into the development process, as practiced in the Lean Startup methodology. When we mention “combining” concepts from these two methodologies, the two aspects in which we are most interested are the frequency with which we have product releases and the frequency with which we gather customer feedback, as represented in Figure 6.4. We believe that shortening both these feedback loops will contribute to the overall value being delivered to all stakeholders and, ultimately, to the success of the project.
6.2.2 Part 1 - Definition of Core Value Propositions

A Value Proposition (VP) is traditionally defined as a statement of the value that a company, product of service offering provides to a customer [31]. We believe this is a useful concept to use when describing the main goals of a project. It can serve as the mission statement for the project and a quick way to explain its rationale. It also serves as a guideline for what kind of features should be included in the system being developed. Only the features that contribute to the delivery of a value proposition should be developed.

We recommend defining a few core VPs for the whole project from the start - this helps in guiding its development and in explaining it to both sponsors and customers. The number of value propositions to define can vary depending on the project; we believe a number between 1 and 3 should be sufficient.

6.2.3 Part 2 - Definition of IS Value aspects

As explained above, one of the main characteristics of this proposal is the inclusion of IS Value as an additional criteria of project success. IS Value is measured as system Use. System Use, in turn, can be defined in a number of ways, depending on the system being developed.

Our goal here is to define the aspects that characterize system Use (and IS Value by extension).

The particular choice of aspects of system use is context dependent. It should be such that enables us to capture the usage of the system as a whole. The Core Value Propositions may also be helpful in choosing these aspects because they represent the high level goals of the system. Examples of aspects of system use include the number of users of the system, the amount of time they spend using the system or the amount of times they perform any specific action that is considered to be representative of system use.
6.2.4 Part 3 - The development cycle

In this section we describe how the bulk of the development work is carried out. Development work is conducted in cycles - the Scrum Sprints. This section explains the layer we add on top of a Scrum Sprint - how it enables us to measure IS value along the development process and use this information to guide development efforts towards increasing this value.

In Figure 6.5 we provide a diagram showing how Scrum Methodology and Lean Startup Methodology can be combined together. The resulting combined methodology is explained in further detail below.

![Diagram](image)

Figure 6.5: The Scrum and Lean Startup methodologies combined.

1. The project starts with a set of *Ideas* that represent the system’s *Value Propositions* to its customers. Those ideas are captured in the *Product Backlog* in the form of *Value Propositions (VPs)*. Items can be added to the Product Backlog at any given time;

*Value Proposition (VP) in this context* The concept of VP was explained before in the context of defining global value propositions that apply to the whole project. Here we use the concept of VP also to name the items in the Product Backlog because we believe this helps in thinking about the connection between these items and the value we expect them to provide to the end customer.

2. The fixed duration (2-4 weeks) sprint cycle starts with a *Sprint Planning* meeting. During this meeting, the team selects a certain amount of top priority *VPs* from the *Product Backlog*, limited by the effort available for each sprint, translates these VPs into MVPs and moves them to the *Sprint Backlog*. An MVP should be defined as a traditional user story in Scrum, with the addition of a list of MVP-specific metrics that should be measured. This metrics can be discussed among the
Scrum team and are ultimately decided on by the Product Owner. This step is further explained in Figure 6.6.

**Minimum Viable Product (MVP)** Some definitions of the MVP describe it as a version of the product that allows the development team to learn the most about the customer with the least effort possible. This definition is, of course, open to interpretation. Sometimes, it may mean it is enough to build fake features (with no functionality or utility for the customer) into the product just to measure the customer's interest in an idea. Other times, however, it may be so that a fully functional product is needed to test an assumption. It may even happen that the effort of building a complete feature is not significantly greater than the effort of building a mock one. These issues must be discussed among the team and ultimately it is up to the product owner to decide on how exactly a VP from the Product Backlog will translate into an MVP on the Sprint Backlog.

---

**Figure 6.6:** The Scrum and Lean Startup methodologies combined.

3. The **Sprint** is where the actual development work is done. During the sprint, we build the selected MVPs together with the mechanisms required to capture the selected metrics for these MVPs (e.g.: a custom-built counter to log clicks on a specific button or an off-the-shelf analytics solutions that can be reused for capturing other metrics to be selected in the future);

4. At the end of a **Sprint** we have a **Shippable Product Increment** together with data that represents customer feedback on the MVPs under development.

5. The **Sprint Review** meeting serves to discuss the previous sprint, including the **Data** from user
feedback. These data will be used to Learn about our assumptions over the Value Propositions. It will be passed as input to the Sprint Planning, where we will decide on whether to further develop specific MVPs or simply to consider the VP complete and remove it from the PB.

6. Back to the Sprint Planning, we now have Data that can help guide our development process.

In summary, the major goal of our proposal is to promote incremental development of features, together with the necessary mechanisms to evaluate the success of those features. More than evaluating the success of previous features, these mechanisms should allow us to Measure and produce Data that will be used to guide following development cycles.

The focus should, in fact, be on measuring relevant data for every feature being developed, ideally. The goal is both to keep development work close to business rationale and at the same time produce Data that will help generate ideas, prioritize features, bug fixes, and every other kind of development work on the system. This may mean changing, expanding or removing previous functionality. Our main goal is that these decisions be based not only on a feeling from the PO, but also on feedback data gathered from real usage patterns.

Figure 6.7 shows the result of a Sprint as both a Product Increment and Data that will serve as input for any following sprints.

Figure 6.7: Each Sprint produces Data to be used as input in subsequent Sprints

33
Chapter 7

Demonstration

This section corresponds to the fourth step in the DSRM – Demonstration. Here we will apply the solution proposed in Section 6.2 (Research Proposal) (the artifact produced by the DSRM process) to an instance of the problem in a real-world organization. In practice, the demonstration will consist of applying the developed methodology to an ISs development project.

The real world IS development project used in this demonstration is called SKAN Platform. The system was developed according to the timeline presented in Figure 7.4. In Section 7.1 we provide the context in which the project emerged and in Section 7.2 we explain how our methodology was used to guide its development.

7.1 The SKAN Platform Information System Project

The SKAN Platform aims to be a platform for knowledge sharing in Agriculture, Food and Forestry sectors across Europe, Africa and Latin America.

In Section 7.1.1 we start by describing the organizational context in which this project exists. In Section 7.1.2 we cover the main high level goals of the project which resulted from initial brainstorming meetings. In Section 7.1.3 we present three technical approaches that were considered for the implementation of the project, along with the chosen one and the reasoning for such choice. Lastly, in Sections 7.1.4 and 7.1.5, we describe the initial project roadmap, the period from project kickoff to the beginning of this demonstration, respectively.
7.1.1 Organizational Context

The *SKAN Platform* initiative has the support of the Portuguese Government and is led by five founding partners:

- Instituto Superior de Agronomia (ISA), Universidade de Lisboa
- Universidade de Évora (UÉvora)
- Universidade de Trás-os-Montes e Alto Douro (UTAD)
- Instituto de Investigação Científica Tropical (IICT)
- Instituto Nacional de Investigação Agrária e Veterinária, I.P. (INIAV)

These founding partners are represented by Inovisa, the entity acting as the project manager for the *SKAN Platform* initiative, as shown in Figure 7.1.

**Inovisa** Born in 2005 inside the School of Agriculture, University of Lisbon, Inovisa was founded with the goal of helping researchers and students in the creation of their own business projects. Inovisa’s main purpose is to promote innovation, entrepreneurship and knowledge sharing in its core operating areas - Agriculture, Food and Forestry - with a strong focus on the Portuguese speaking countries community.

As the project manager for *SKAN Platform*, Inovisa represents the interests of all major stakeholder organizations in this project. Inovisa was the de facto client for this development effort and the research team had no other contact with the remaining organizations. A simplified Organization Breakdown Structure is provided in Figure 7.1

![Figure 7.1: Simplified Organization Breakdown Structure (OBS) for the *SKAN Platform* project](image-url)
7.1.2 Main Goals

Initial discussion on the SKAN Platform project began in February 2014 with brainstorming meetings on the project goals and possible approaches. During these meetings it was established that the project’s main high level goals would be:

- To enable knowledge sharing inside Agriculture, Food and Forestry communities in each of the selected countries;
- To also enable knowledge sharing between these communities and across different countries;
- To create a platform where people in these sectors can get to know and communicate with each other.

7.1.3 Technical Approach

Technical approaches considered Given the goals described above, several technical approaches were considered for the implementation of the project.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing social networking platforms (e.g. Facebook or LinkedIn)</td>
<td>Simplicity and non-existent/little software development effort.</td>
<td>Lack of brand presence (no custom domain name, no custom website branding); Lack of flexibility in communication between system administrators and customers (e.g. impossibility of sending custom newsletters); Lack of flexibility for further development of the system and adding new features.</td>
</tr>
<tr>
<td>COTS software</td>
<td>Template solutions have the potential to greatly increase development speeds.</td>
<td>Lack of flexibility; lack of familiarity with specific products by the development team.</td>
</tr>
<tr>
<td>Custom software</td>
<td>Flexibility.</td>
<td>High development effort.</td>
</tr>
</tbody>
</table>

Table 7.1: Technical approaches considered for SKAN Platform implementation

After considering the advantages and disadvantages summarized in Table 7.1, the Project Manager found the best solution to be building custom software. Given extremely uncertain starting requirements, the Project Manager favored this approach for the flexibility it provided.

The Ruby On Rails framework Having decided to go with the route of building custom software, we chose Ruby on Rails (also known as just Rails or RoR) as the basis for the software development work. Ruby on Rails is a web application framework written in the Ruby programming language. It is designed specifically to make building web applications easier [32, 33]. Rails is praised for its simplicity and the
productivity increases it seems to enable [34]. It targets the development of software very similar to what we intended to develop here and despite the lack of familiarity with the technology, the team decided it would be a good fit for the project.

7.1.4 Project Roadmap

After initial meetings, it was established that the project would start by focusing on the Portuguese (PT) community for stage 1 - we would start developing a system with scalability to other countries in mind, but targeting the Portuguese community in its first iteration. In stages 2 and 3 the solution would then be extended to include other communities, represented in Figure 7.2. From here onwards, these country communities will be called country networks, or simply networks, which was the terminology adopted by the project team.

In stage 2, we would create the Mozambican (MZ) community. In stage 3, we would create the remaining planned communities for Angola (AO), Brazil (BR) and Cape Verde (CV).

Stage 1. Create a minimal system to be deployed and tested in the Portuguese community. This system would be materialized in a website and available to the public. Once live this system would replace an existing standalone Portuguese initiative named Rede INOVAR which had goals similar to those of SKAN and was controlled by the same founding partners. An important aspect to note here is that the solution devised for one network should be similar and easy to replicate for other networks. As we move from one stage to another and include more and more networks, each network is presented in a similar fashion and the features for each evolve as whole.

Stage 2. The main goal in stage 2 was to design a system capable of extending the solution created in stage 1 to other country networks. This system would be a website that would somehow aggregate the content and articulate the relation between the different networks as part of a bigger initiative, the SKAN Platform initiative, as represented in Figure 7.2. The most important aspect of stage 2 was to create a scalable system that allowed a number N of other networks to be easily created in the future, while maintaining presentation consistency in the SKAN Platform website,
the aggregator of all country networks. Figure 7.3 helps understand this relation between SKAN Platform, the aggregator, and its child networks. Stage 2 would end with the creation of a second network - the Mozambican (MZ) network.

Stage 3. Finally, in stage 3, the solution would be extended to the remaining planned networks - Angola (AO), Brazil (BR) and Cape Verde (CV).

![Figure 7.3: Identical standalone networks are aggregated in SKAN Platform](image)

7.1.5 From Project Kickoff to Demonstration Start

As described in the beginning of this chapter, project kickoff took place in February 2014. This initial period served mainly to brainstorm ideas, discuss project vision, get first contact with new technologies to be used in the project and develop internal prototypes. During these first meetings the number of participants in the meetings, on the client side, was variable and so were their roles, responsibility and autonomy.

During this period we were able to better understand some important features of the project that made it a very good candidate for the demonstration of the Proposal in Chapter 6, namely:

- Flexible scope;
- Non-life-critical;
- Inexperienced product owner;
- Fuzzy requirements.

In October 2014 we started the demonstration of the proposal presented in Chapter 6. At this point,
in terms of the roadmap presented in Figure 7.2, we were still in Stage 1. We had created a minimal website centered on sharing of user-generated content but this solution was not yet suitable for transition to Stage 2.

The SKAN Platform IS project is a very interesting problem in which to demonstrate the solution we propose because although there is a defined project vision, restrictions on requirements are few and there is a clear need for guidance in the development process.

“What system should we build?” and “What should we build first?” were question asked since the beginning of the project.

In Section 7.2 we explain how we applied the methodology proposed in order to help answering these questions and improve the development process.

7.2 Demonstration

Figure 7.4: SKAN Platform project timeline

In October 2014 we started applying the methodology proposed in Chapter 6 thus beginning its demonstration in a real world scenario. This meant not only a change in the development methodology used until then but also some changes in team organization, communication and roles. We start by describing the team structure in Section 7.2.1. In Section 7.2.4 we explain how our methodology was applied to this project and how it was used to guide the development of the SKAN Platform IS.

7.2.1 Team Structure

In Figure 7.5 we present the team structure for the SKAN Platform project. For each member we present their Scrum role. For members in the Development Team we also provide a brief description of their core functions in the project.

As explained in Section 6.2, we can see in the Development Team multiple types of development work
being performed.

Whereas in the beginning of the project, marketing-related work was being performed outside the Scrum Team and outside the scope of this demonstration, shortly after the beginning of the demonstration, a new Team Member, focused on this type of work, was included in the Development Team.

Having had the opportunity to observe the team before and after this event, we could verify that this was a very positive change. It helped bring together software development aspects and business aspects, contributing to a more holistic perspective on the system being developed.

![Figure 7.5: Team Structure](image)

7.2.2 Definition of Core Value Propositions

From the goals presented in Section 7.1.2, we can now derive more concrete Value Propositions for the SKAN Platform project:

- **VP1**: *SKAN Platform will provide a place where customers can discover and share news, events, and various other content related to Agriculture, Food and Forestry areas.*

  Existing solutions offering similar value propositions to the targeted market (Agriculture, Food and Forestry) consist mainly of personal blogs or websites where editorial content is managed by the owner entity. SKAN Platform, by contrast, will rely on user generated content, serving as a place to gather content from multiple disparate sources.

- **VP2**: *SKAN Platform will provide the means for its customers to discover and communicate with their peers.*

  Existing solutions such as LinkedIn can be too general. The goal here is to provide a means for customers to easily understand who their professional peers are, both in their countries and
abroad, and a way for them to contact each other, bringing the community closer and opening way for future partnerships.

7.2.3 Definition of IS Value aspects

As explained in Section 6.2, the aspects that define IS value can vary substantially depending on context. Here we have chosen to use four base metrics to characterize IS Value. We present them below together with the respective measurements taken for the duration of this demonstration.

**Active Users per Month.** Active users in a given month are users who have had at least one session in the website during that month. A session is defined as one or more interactions with the website and ends after 30 minutes of inactivity. It is important to note that a session does not imply a login to the website. Customers can use certain parts of the website without having created an account or having performed a login; these interactions count as sessions.

In Figure 7.6 we present the number of active users per month, between October 1, 2014 and August 31, 2015.

![Figure 7.6: Active users between Oct 1, 2014 and Aug 31, 2015](image)

**Webpages Visited per Month.** Webpages visited per month is the total number of webpages visited each month, including repeated views of the same page.

In Figure 7.7 we present the number of webpages visited per month, between October 1, 2014 and August 31, 2015.

**Average Time on Page.** Average time on page represents the average time a user spends on a given webpage.

In Figure 7.8 we present the average time on page per user per month, between October 1, 2014 and August 31, 2015. Between April and March we see an clear increase in the average time spent per page. This could have been a side effect of the temporary decrease in Active Users and Pageviews during this period but, looking at the following months, we see a new increase in those metrics and a continuation
of the trend in Average Time on Page. An alternative, and more likely, explanation is that this period saw an increase in the quantity and quality of the content available on the SKAN Platform which can explain an increased engagement from the users.

New Registered Users per Month. Registered users are users who have an account in the SKAN Platform. Users can easily and freely create an account on the website which gives them access to features not available to non-loggedin users such as the user list, content creation, commenting, newsletter, among others.

In Figure 7.9 we show the number of new registered users per month, between October 1, 2014 and August 31, 2015.

It is important to note that the SKAN Platform IS can also provide value for users without an account and, in contrast with more profit-oriented products, this is inline with the project's sponsors’ goals. This is why it is relevant to look both at New Registered Users and also at Active Users, who do not have an account, as overall success indicators.

Figure 7.10 is based on the same data supporting Figure 7.9 but shows the cumulative evolution of registered users per month. At the end of this demonstration the system has 318 registered users.
7.2.4 Development Work

While in previous sections we described more high level characteristics of the project such as Team Structure or the process of defining Global Success Metrics, in this section we focus on the actual development work throughout the project.

Here we will cover the core of our methodology, as presented in Figure 6.5 from Chapter 6. We will select some of the most relevant development tasks from the 6 feature domains presented below and explain how they went through the development cycle described in Figure 6.5. More specifically, for each of these tasks, we will explain:

- The overall story of the task - how it progressed from Idea, to Product Backlog Item, to Sprint Backlog Item;
- What was built in the Build Activity (this may differ from the feature description; e.g. when the team decides to implement an MVP instead of the fully functional feature) How the Build and Measure actions were implemented during the Sprint;
- What metrics were measured in the Measure Activity. (It should be noted that the Measure and Learn Activities for a particular feature do not happen in the same sprint; first we build the feature
and the mechanisms to that will allow us to keep track of the proposed metrics; then, in subsequent sprints, we collect the measurements in the Measure Activity of that Sprint.

- How the resulting Data from the Measure action contributed to Learning and this influenced subsequent development work.

**Feature Domains**  The system functionality can be roughly organized into 6 domains:

- Users
- Posts
- Events
- Comments
- Messaging
- Notifications

**Task 1 - Profiles and search (Users)** Creating user profiles and the ability to search for specific users, was one of the first development tasks of this project. These features are directly related with value proposition **VP2** (Section 7.2.2). Table 7.2 provides the details for this task.

<table>
<thead>
<tr>
<th>Task</th>
<th>Task 1 - Profiles and search (Users)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>As a user, I want to have a public profile and be able to search other users' profiles so that my peers can find me and know more about me and also me about them.</td>
</tr>
<tr>
<td><strong>Build Activity</strong></td>
<td>Create user profiles; create a listing of users and provide a search mechanism to lookup users.</td>
</tr>
<tr>
<td><strong>Measure Activity</strong></td>
<td>Number of user searches performed; Number of user profile views; Number of users list views.</td>
</tr>
<tr>
<td><strong>Learn Activity</strong></td>
<td>As we can see in Figure 7.11, there is a spike in users listing views and profile views, around the time the system started to be discussed with the public in general. Searches for specific users have remained comparatively low; this may indicate that user profile views are due to posts and comments published by those users.</td>
</tr>
<tr>
<td><strong>Added to Sprint Backlog</strong></td>
<td>Before Oct 2014</td>
</tr>
</tbody>
</table>

Table 7.2: Task 1 details - Profiles and search (Users)

**Task 2 - Upvotes and downvotes (Posts/Comments)** Having upvotes and downvotes for posts and comments was a feature discussed early on during the initial brainstorming for the project. The motivation for this feature was twofold: on one hand, it should increase user engagement and on the other hand, information about upvotes and downvotes could be used to improve the way content was displayed on pages (e.g. giving priority to highly rated content). Table 7.3 provides the details for this task.
Table 7.3: Task 2 details - Upvotes and downvotes (Posts/Comments)

Task 3 - Messaging between users MVP (Messaging) This feature relates to direct communication between users. It was part of what the Product Owner saw as core functionality of the system. Nevertheless, after considering development effort and uncertainty about user interest in the feature, we decided to build a landing MVP for this feature. New sections and links would be created on the website but without really implementing the messaging functionality. We would present a feedback form to users and also measure clicks in relevant links in order to gauge user interest in the feature. Table 7.4 provides the details for this task.

46
Table 7.4: Task 3 details - Messaging between users MVP (Messaging)

<table>
<thead>
<tr>
<th>Task</th>
<th>Task 3 - Messaging between users MVP (Messaging)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>As a user, I want to be able to send and receive messages from other users, so I can communicate with other peers using the platform.</td>
</tr>
<tr>
<td>Build Activity</td>
<td>Landing page type MVP or non-functional MVP. New sections and links pointing to a feedback form. This feedback form asked users whether they saw value in having a messaging system inside the platform or if they preferred relying on social networking information provided by the users in their profiles in order to contact them (e.g. links to personal LinkedIn and Facebook pages)</td>
</tr>
<tr>
<td>Measure Activity</td>
<td>Number of clicks in &quot;Inbox&quot; and &quot;Send Message&quot; links; Form responses.</td>
</tr>
<tr>
<td>Learn Activity</td>
<td>Discussion on fully implementing this feature happened again between 2 and 3 sprints after current sprint. The data we measured, however, showed 204 link clicks but no answers to the form. We decided this represented a lack of user interest in the feature and postponed its implementation indefinitely, to be reevaluated later.</td>
</tr>
<tr>
<td>Added to Sprint Backlog</td>
<td>2014/12/01</td>
</tr>
</tbody>
</table>

Task 4 - Content creation (Posts) This task represents an interesting aspect of this development methodology since it does not include new software development and because it was motivated by
analysis of data generated by following metrics defined in previous sprints. More specifically, this task consisted of content creation by the development team, after observing low average times on page and low quantity of content created by users as shown in Figure 7.14. Table 7.5 provides the details for this task.

<table>
<thead>
<tr>
<th>Task</th>
<th>Task 4 - Content creation (Posts)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>As a user, I want so see activity on the website in the form of up to date news and events so that I have a reason to spend time on it and create content myself.</td>
</tr>
<tr>
<td><strong>Build Activity</strong></td>
<td>Gather content from multiple sources of news in Agriculture, Food and Forestry and create new posts in the platform.</td>
</tr>
<tr>
<td><strong>Measure Activity</strong></td>
<td>Total number of posts published (including by team members); Number of posts published by development team members; Avg. Time Spent on page.</td>
</tr>
<tr>
<td><strong>Learn Activity</strong></td>
<td>Hoping to increase user activity, the team started also publishing content to platform and this strategy seems to have worked. The results are modest but we can clearly see in Figure 7.14 a positive correlation between content generate by team members and content generated by users. Furthermore, together with the increase in the number of posts, we also see an increase in average time spent on page - which have remained steady ever since.</td>
</tr>
<tr>
<td><strong>Added to Sprint Backlog</strong></td>
<td>2015/02/05</td>
</tr>
</tbody>
</table>

Table 7.5: Task 4 details - Content creation (Posts)

Figure 7.14: Posts by development team members, total number of posts (including by users) and average time on page between Oct 1, 2014 and Aug 31, 2015 (Avg. Time on Page plotted on secondary axis on the right)

**Task 5 - Calendar view (Events)** Table 7.6 provides the details for this task.

**Task 6 - Messaging between users (Messaging)** This task corresponds to the actual implementation of the messaging functionality. Task 3 dealt with the same user story; however, at the time, considering, among other things, the amount of effort needed to implement the feature and the amount of active users measured, the Product Owner had decided to postpone it. At the time, the Product Owner had decided there were more prioritary tasks, such as improving the existing experience and bringing more users to
Task 5 - Calendar view (Events)

**Description**
As a user, I want to be able to view future events not only as a list of items but also in calendar view, so that I can get a big picture view of events happening in each month, week or day.

**Build Activity**
Create a calendar view for events.

**Measure Activity**
Events listing pageviews; Calendar pageviews;

**Learn Activity**
Figure 7.15 shows the data produced in the measure activity.

**Added to Sprint Backlog**
2015-03-28

Table 7.6: Task 5 details - Calendar view (Events)

![Events Index Listing VS Calendar View](image)

Figure 7.15: Events index/list view compared to calendar view between Oct 1, 2014 and Aug 31, 2015

The website. Now, in task 6, we deal with the actual implementation of the messaging functionality. Table 7.7 provides the details for this task.

Task 6 - Messaging between users (Messaging)

**Domain**
Messages

**Description**
As a user, I want to be able to send and receive messages from other users, so I can communicate with other peers using the platform.

**Build Activity**
Create a message inbox for each user; allow each user to send messages to any other user in the platform, either via a direct link in each user profile or by selecting a user in a form for new messages.

**Measure Activity**
New messages created; inbox views and opened messages.

**Learn Activity**
Figure 7.16 shows the data produced in the measure activity.

**Added to Sprint Backlog**
2015/03/13

Table 7.7: Task 6 details - Messaging between users (Messaging)

Task 7 - Notify users of important events (Notifications)
The main motivation for this task was to promote user engagement by notifying users of important events in the platform, such as upvotes and downvotes on a user's posts and comments; new comments on a user's posts and new messages in a
Figure 7.16: Messages created VS pageviews on inbox and opened messages between Oct 1, 2014 and Aug 31, 2015 (Pageviews plotted on secondary axis on the right)

user’s inbox. Table ?? provides the details for this task.

<table>
<thead>
<tr>
<th>Task</th>
<th>Task 7 - Notify users of important events (Notifications)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>As a user, I want to receive notifications of important events in the platform, so I know when someone interacts with me.</td>
</tr>
<tr>
<td><strong>Build Activity</strong></td>
<td>Create a notification mechanism with visual representation in the website, indicating the number of new notifications for new messages, new comments on a user’s posts and new upvotes and downvotes on a user’s posts and comments.</td>
</tr>
<tr>
<td><strong>Measure Activity</strong></td>
<td>New messages created; inbox views and opened messages.</td>
</tr>
<tr>
<td><strong>Learn Activity</strong></td>
<td>Figure 7.17 shows the data produced in the measure activity.</td>
</tr>
<tr>
<td><strong>Added to Sprint Backlog</strong></td>
<td>2015/04/13</td>
</tr>
</tbody>
</table>

Table 7.8: Task 7 details - Notify users of important events (Notifications)

Figure 7.17: Notifications generated VS pageviews on notifications dashboard between Oct 1, 2014 and Aug 31, 2015 (Pageviews plotted on secondary axis on the right)

**Task 8 - Weekly newsletter (Newsletter)** Once again, this task is intended to increase user engagement by sending them a weekly newsletter with a summary of relevant news and events. Table 7.9 provides the details for this task.
End of the development period  By the end of the development period, we had built an information system capable of supporting knowledge sharing and peer discovery for the Agriculture, Food and Forestry sectors across communities in Portugal, Mozambique, Angola, Brazil and Cape Verde.

This resulting information system is supported by a website, reachable in the skanplatform.org domain. As of the end of August 2015, the system serves 318 registered users and a total of 1139 active monthly users, after a peak of 2474 two months before. Figure 7.19 showcases the website’s homepage; further details and screenshots can be found in Annex A.

The project sponsors are happy with the results of the development work but recognize there is still work to be done, namely in improving the quality and quantity of the content delivered by the system and in promoting user engagement across the different country communities. While this research team will no longer be actively involved in further development of the system, the client is in position to continue its improvement, especially in the areas mention above. The client, who was part of development effort, is aware of the mechanisms implemented to gather user feedback, and will hopefully use them in order to increase the value provided to their customers - the information system’s customers.

In Chapter 8 we evaluate the performance of the methodology described in the present chapter.
Figure 7.19: Skan Platform website’s homepage screenshot, as of 18 September 2015
Chapter 8

Evaluation

This section deals with the Evaluation step in the DSRM. Here we reflect on the results presented in the previous chapter and compare them with our objectives in order to understand how well this solution addresses the problem described in Chapter 3.

This chapter is divided into two sections, corresponding to two evaluation methods - the first proposed by Hevner et al. [35] as an evaluation method for design science research; the second, based on the research principles for design-oriented information systems research, advanced by Osterle et al. [36]. Below we provide a description of each of these two methods and how they were used in this evaluation. Sections 8.1 and 8.2 present the results of their application in this research.

**DSR evaluation methods (Hevner et al. [35])** Hevner et al. propose five methods that can be used to evaluate design artifacts:

- **Observational** The artifact is studied and evaluated in a business environment; case studies or field studies can be employed.

- **Analytical** The artifact is examined by itself and evaluated in terms of its characteristics, such as complexity or performance.

- **Experimental** The artifact is evaluated in terms of specific qualities (e.g. usability) through experiments in controlled environments or simulations.

- **Testing** The artifact is evaluated in terms of how it interacts with other systems, i.e. in terms of its interfaces or in terms of internal tests (e.g. execution paths in the code)

- **Descriptive** The artifact is evaluated through an informed argument, using information from the knowledge base (e.g. existing research), that should build a convincing case for the artifact’s utility.
From these five methods, the first and the last emerged as the most adequate for the solution we proposed.

The Observational method relates to the work presented in Chapter 7 - Demonstration. As Hevner et al. [35] describe, one of the ways in which this method can be accomplished is through the use of a case study, which is defined as the study of the artifact in a business environment. In the previous chapter we were focused on describing the process of using the artifact in a real world scenario. In the current chapter, however, we adopt a more critical stance and try to evaluate the artifact's performance; we will discuss the empirical evidence and how it relates to what was expected. We will also provide the results of interviews with the participants in the case study, with other practitioners and with a person from academia who also has extensive in IS project management. This analysis is presented in Section 8.1.

The Descriptive method is essentially a combination of what we have presented in Chapter 4 - Related Work and Chapter 6 - Proposal. By studying existing solutions for this problem and suggesting a new approach, we have built an argument based on existing research for the usefulness of the solution we propose.

**Design-oriented IS research principles (Osterle et al. [36])** Osterle et al. suggest four basic principles that design-oriented IS research must comply with. These principles are:

- **Abstraction** The artifact must be applicable to a class of problems;
- **Originality** The artifact must contribute to advance the body of knowledge;
- **Justification** The artifact must be justified in a clear manner and must allow for its validation;
- **Benefit** The artifact must yield benefit for its stakeholders, either immediately or in the future.

We can then evaluate our solution, or artifact, in terms of how it complies with these principles. This analysis is presented in Section 8.2.

### 8.1 Empirical observations and interviews (Hevner’s Observational method)

In the context of this research, Hevner’s Observational method consists of evaluating the demonstration of our proposal. This demonstration, described in the previous chapter, consisted of the use of our proposal as a methodology for development of an information system, the SKAN Platform.

In Section 8.1.1 we discuss observations gathered during the demonstration and in Section 8.1.2 we collect feedback on the participants in the demonstration in order to evaluate their perspective on the
merits of our proposal.

8.1.1 Empirical observations

Definition of core Value Propositions As depicted in the project timeline in Figure 7.4, initial discussion and development on this project started before the beginning of the demonstration of our proposal. During this time, development work was already being carried out alongside discussions on the project direction and goals. The lack of a clearly defined vision for the project meant that for almost every new feature proposed, a whole new discussion on the project’s purpose would arise each time. Most of these discussions where focused on the potential benefits of each feature and lacking a unifying vision for the project, every suggestion of a new feature would spark discussion on a different possible direction for the project.

The beginning of the demonstration and the definition of project goals, translated into Value Propositions, provided a useful vision for the project. We decided we would focus on: a) providing a place where customers could discover and share news, events and other content relevant to the target audience; and b) providing a means for customers to discover and communicate with their peers.

Keeping these Value Propositions in mind helped us understand how value was being created for the customer and what features would be the most useful in implementing these propositions and increasing this value.

Definition of IS Value aspects In the proposal presented in Chapter 6 we argued that system $Use$ can represent a measure of IS Value and that we need to specify, for any given project, the aspects of the system that will represent system $Use$.

In this demonstration, we chose four aspects to characterize system $Use$: The number of active users per month (with or without a user account in the system), the number of webpages visited per month, the average time on page and the number of new registered users per month.

It is debatable whether these aspects were the best choice but we think they were able to reflect system $Use$ and convey a sense of the value being delivered by the IS. More importantly, these aspects were being measured continuously alongside development work and proved useful for guiding development effort. An example of such an instance was the realization that at some point we were trying to test user interest in some features but we had only a very small amount of active users on the system. This prompted the team to guide efforts towards increasing the number of users on the platform and shortly after we could see the results as the number of active users increased and the average time spent on page doubled.

An interesting question to ask about the IS Value aspects is how many we should choose and how to choose them. The answer depends heavily on the project under consideration. For instance, in this
demonstration, we could have chosen the number of posts created each month as another IS Value Aspect, and one that we could easily relate to the value proposition of creating and sharing content.

In practice this was not a problem because of how the development cycle works in our proposal. In the development cycle, new features are developed together with mechanisms to measure their individual value (a feature-specific version of system Use). In practice we noted that this information was similar to the global IS Value Aspects and could be used in the same way. This meant that, in the particular case of the number of new posts per month, once this feature was implemented, we could start considering it as an IS Value Aspect that not only is used to guide development, but also contributes to the final appraisal of IS Value.

Development work In our proposal the development work is spread in development cycles that correspond to a Scrum Sprint with an additional layer for tracking the value of the features being developed and for guiding the development efforts in a way that increases the value of the system as a whole.

The proposal states that when a Value Proposition item is picked from the Product Backlog and placed in Sprint Backlog, it must be described as a user story, together with the metrics that will be used to measure the value of that new feature. We call an item in the Sprint Backlog an MVP because it may or may not represent a complete functional feature; it is up to the Product Owner to decide whether to fully implement a feature right away or to first test the potential value of the feature by implementing a mock version of it.

Overall, we found it extremely useful to build each feature together with mechanisms to provide continuous feedback on the value that it is delivering to customers. First, thinking about what will be measured helps in defining the feature itself because it forces us to think how it will provide value. Second, the information collected on each feature provided extremely useful feedback that enabled us to better understand the impact of these features on the system as a whole and based on these data, make decisions about where to allocate development effort.

Examples of ways in which this information was used include the decision to allocate efforts on other areas instead of improving the search feature because almost nobody was using this feature; the decision of allocating effort to marketing and content creation because user activity was very low (and the ability to measure significant results of this action); the understanding that a downvoting feature was almost useless without anonymity; the impact of the user newsletter in user engagement.

One way in which we had expected a different outcome was in the use of the concept of MVP in the Sprint Backlog. Although the complexity of each MVP in the product backlog is ultimately decided by the Product Owner, we expected to make greater use of mock features to test the potential value of these features before committing to bigger development efforts. A true mock feature, completely non-functional, was employed only once during the project, for an early experiment with the messaging feature (Task 3 in Chapter 7 - Demonstration).
We believe this happened mainly due to two reasons: The first has to do with the low amount of effort that was required to implement a simple version of most features; most of the time it was easier to implement a minimally functional feature right away and improve it later based on the value inferred from the metrics collected for that feature. The second reason for the scarce usage of mock features was that implementing a non-functional feature for testing purposes with users proved a strange concept for the Product Owner and other members of the team; they understood the goal and agreed to do it but it is understandable that some reservations exist when exposing the customer to this practice, especially when it is done for the first time.

In summary, we found great benefits in trying to measure the value of each individual feature. The practice had a great impact in better understanding the system and in guiding its development. In addition, the data collected for each feature can also be used in the final appraisal of IS Value and ultimately, of the success of an IS development project.

**IS Value as a dimension of project success** One of the main aspects of this proposal is the inclusion of IS Value as an additional way to measure the success of IS development projects.

We think this demonstration showed that IS value is useful as a dimension of IS project success. By using IS Value as a project success dimension, we provide the team with a meaningful aspect they can strive to optimize, in addition to optimizing for cost, time or scope.

The perspective of the team on the importance of IS Value as a project success dimension is presented in the section below.

### 8.1.2 Interviews

In order to better evaluate our proposal, we conducted interviews with four different types of audience. The first type were the members of the project team, from Inovisa, who were part of the development process. These interviews were made through surveys, which are presented below. The goal was to evaluate both the rationale for the proposal and the actual practice of using the methodology. Three participants in the project were interviewed.

For the second and third types of audience, we used a simplified version of the same survey, removing questions related to the experience using the methodology since these people were not part of the development process. The second type refers to the interview conducted with the project sponsor, whose feedback has a special weight. The third type refers to other practitioners inside Inovisa, who managed or took part in similar projects in the past and who are familiar with the SKAN Platform. Four people were interviewed.

Finally, the fourth type of audience was someone from academia who also has extensive professional experience managing IS development projects - Professora Maria do Rosario Ponces de Carvalho. This
was an open, semi-structured interview with the goal of collecting her feedback in terms of validating both the problem and the proposal.

In total, 9 people provided their feedback.

Below is the survey used for the type 1 interviews. Types 2 and 3 used a subset of this survey composed by questions 1, 2, 4, 5, 9.

1. Do you agree that the value provided by an Information System should be part of the criteria for evaluating the success of an Information System development project, alongside the cost, duration and scope of the project? (1: Completely disagree, 10: Completely agree) * 1

2. Do you agree that the amount of use of the system (e.g. visits to the website) as a form of feedback from the users is a good measure of the value provided by an Information System? (1: Completely Disagree, 10: completely agree) *

3. Do you agree the metrics we measured during the process of developing the SKAN Platform (e.g. the number of registered users on the platform or the time they spent on the website) helped you prioritize development efforts (e.g. choosing which feature to develop first or where to focus marketing efforts)? (1: Completely disagree, 10: Completely agree)

4. Sometimes launching incomplete features and testing them with real users allows for faster learning about their preferences so that development effort is not wasted on features that users do not value. How do you feel about launching incomplete features for testing with users? (1: Very uncomfortable, 10: Very comfortable) *

5. How do you evaluate the success of the SKAN Platform development project in terms of the value delivered by the system to its users? (1: Very unsuccessful, 10: Very successful) *

6. Do you agree that the use of this IS development methodology contributed to the success of the project? (1: Completely disagree, 10: Completely agree)

7. How do you classify your overall satisfaction with the development methodology used to develop the SKAN platform? (1: Very dissatisfied, 10: Very satisfied)

8. Would you consider using this methodology again in the future? (1: Definitely not, 10: Definitely yes)

9. How do you evaluate the overall success of the SKAN Platform development project? (1: Very unsuccessful, 10: Very successful) *

**Interview results - project team members** From the results in Figure 8.1, we see that in general, project team members agree with the basic premise of this proposal, which is that IS Value should be

---

1 Questions marked with an asterisk represent the subset used for type 2 (sponsor) and 3 (other practitioners) interviews.
one of the criteria used to assess the success of an IS development project, alongside traditional criteria of time, cost and scope.

They also agree that using system use as a way to measure the value of specific features was an effective way to prioritize development efforts or, in other words, to guide the development process. This was another fundamental premise of this proposal. Using MVPs with varying levels of functionality and showing them to users does not seem to pose a problem.

In terms of the overall value of the system, however, we see that team members do not consider the project to have been very successful. From all the conversations during the development process, we believe this is mainly due to the project's scope. There were high expectations for the project's scope in the beginning of the project and some of them could not be met due to time constraints. The development team was small and the project progressed slower than what the project sponsor and team expected in the beginning. The lower scores in these interviews were from a member who was not continuously involved in the project but only during specific periods. Nevertheless, her feedback is important because the scores are also lower in the more generic questions 1 through 4. We can also see that this person is somewhat uncomfortable with releasing partially complete MVPs to customers, which is a position not shared by any of the other respondents.

Despite this, we see that in general, the members of the project team found value in the methodology we proposed. They think it helped in guiding the development process and contributing to the success of the project, they were overall satisfied with using it and would use it again in the future.

Figure 8.1: Interview answers from the project team members. Different colors denote different members.
Interview results - project sponsor  The project’s sponsor also validates the main premises in our proposal as shown in Figure 8.2. He strongly believes that the value provided by an IS should be part of the criteria used for assessing its success. He also believes that system use (of which website visits are an example) is a good proxy for measuring IS Value. He is also very comfortable with deploying incomplete features in order to learn faster about customer preferences. In terms of overall project success and value delivered to users, his evaluation lies in the middle of the scale, similar to the opinions of the project team and very likely for the same reasons.

Interview results - other practitioners  Looking at the first questions (1, 2 and 4) and the results in Figure 8.3, we see that practitioners also strongly validate the fundamental premises of our proposal. They agree that IS Value should be a criteria in evaluating the project success and they agree that system use is a good proxy for measuring IS Value.

As outsiders to the project development, their views on the value delivered by this system in particular and the overall success of the project, are better than those of the project team and sponsor as shown in Figure 8.3.

Interview results - member from academia with extensive professional experience in IS project management  Maria Ponces de Carvalho holds a teaching position at Instituto Superior Tecnico and has extensive professional experience managing IS development projects. We sought her feedback on this proposal because she is in a position that allows her to appreciate both the theoretical aspects and also the needs of real world professional environments.

In terms of the problem definition, she recognizes that using the traditional project management triangle or iron triangle of cost, time and scope, is a limited way to assess project success. She recognizes
that the value of the system being developed is usually not explicitly taken into account and prevalent methodologies lack mechanisms to increase it.

She notes that many systems are not developed in small increments that are continuously deployed to the target audience of system users. But in the cases where this is a possibility, then our proposal can be very valuable. Both because it provides guidance in the development process, and also because it provides the organization with a measure of the value the system has for its customers, which is invaluable information from the organization's perspective.

She agrees with the idea of continuously measuring the value of specific features as a way to guide the development process and she thinks system use can be a good proxy for measuring the value of the IS. Here she reminds us that it is important to make clear the distinction between value for users and value for the organization. And indeed what we are measuring with the use of the concept of IS Value in this research is the value provided to customers. In terms of the value provided to the organization, we can guess that value provided to customers will translate into value created for the organization but this is nothing more than a suggestion.

In summary, provided that a specific project can be developed in a way that is compatible with this proposal, she agrees that this is an interesting way to tackle the problem that was presented, to increase the value delivered to customers and to generate important feedback for the organization.

**Summary** These interviews were a great instrument in evaluating our proposal. From the results collected here, we can say that the basic premises of our proposal were validated. Namely:

- That IS Value should be part of the criteria used for assessing the success of IS projects, in addition
to time, cost and scope;

- That system use is a good proxy for measuring IS Value;
- That IS value is useful to prioritize and guide the development effort;
- That the methodology we propose is effective in using IS Value to guide the development process.

Then, there’s the question of whether or not this methodology lead to the development of a system that is considered to be successful. The feedback in terms of value delivered to users and overall project success, points to less than ideal outcomes in these aspects. Part of this may be due to high initial expectations in terms of project scope that were not matched. The other part may be due to the very own IS Value measurements that are now available - in other words, the number of people using the system and how they are using it. Having this information available means the project team and sponsor can now evaluate the project along a new dimension - IS Value. On the other side, it also means that it may be easier to take corrective actions that do not necessarily imply the development of new software.

The expert feedback on this proposal was very positive. On one hand, it recognized the increased value to customers through delivery of better information systems. On the other hand, it noted the value delivered to organizations in the form of feedback on the systems they develop.

### 8.2 Discussion on the principles for IS research (Osterle et al.)

**Abstraction**  The methodology we propose is an information systems development methodology - it is therefore not targeted at a specific problem but at information systems in general. Amid multiple alternatives, however, it will not be the best fit to all types of problems. It is our opinion that the methodology we propose can be applied to at least the types of projects already using agile methodologies, with an emphasis on those with open scope and high uncertainty in requirements.

**Originality**  The practice of selecting and following usage metrics in itself is a widely adopted one - from measuring traditional sales results to user engagement and using these data to inform the business development process. What is different in our proposal is the formal inclusion of this practice in the development of the information system, guiding the development while the system is being built, shortening the feedback loop every traditional business must go through. During the demonstration of our proposal we were able to empirically observe the shortening of this cycle: With little information and experience in similar projects, we noted a tendency for 'feature creep' from the product owner; when presented with data, however, he was able to make decisions quicker and drop less priority features. Bringing the feedback from after the final release to something that is available during the whole project and for the specific features being developed is what shortens the feedback loop and allows less effort to be wasted in the process of creating value for stakeholders. Including this aspect in the development
methodology is what confers originality to our proposal.

**Justification**  Chapter 4 - Related Work, covers the justification for the artifact. In summary, the artifact tackles the very important task of developing information systems. More specifically, those information systems where the biggest challenge is the discovery of a solution in dynamic and fast-changing environments, as opposed to execution-focused projects where extensive upfront planning is possible and desirable. The former are becoming more and more prevalent as organizations strive for competitive advantage by exploring the possibilities offered by technology, many times with unclear purposes and requirements. Traditional information systems development methodologies are not suited for this type of problem; Agile methodologies bring flexibility and seem to be an improvement over traditional approaches; our proposal builds on these efforts and targets the problem of guiding the efforts of the development process in this dynamic environment.

**Benefit**  As described before, the main benefit of our proposal is that it helps reducing wasted effort in the development process by focusing on the needs of the users of the information system under development. This shortened feedback loop helps to maximize resource efficiency from the perspective of the project sponsor and to create value sooner, from the perspective of final customer and user of the resulting information system.
Chapter 9

Conclusion

9.1 Main Contributions

Software-based Information Systems are increasingly pervasive in our lives. Decades ago they started as a way of streamlining operations but today they represent the core component of many products offered to the end customer. This means the challenges of developing information systems are also changing. While before we were automating and optimizing existing processes, we are now creating entirely new software-based products designed to cater to an ever increasing range of customer needs.

Traditional software development methodologies, such as Waterfall, best suited for well defined requirements or life critical projects where extensive planning is possible or desirable, are no longer fit for many of today’s challenges. They seem a less than optimal strategy and fail to leverage the flexibility software provides.

Agile methodologies, on the other hand, are more adequate for dynamic environments and bring flexibility to the development process.

What we have proposed here is a step further in exploring the Agile approach. We proposed a methodology that is based on Scrum, uses concepts from the Lean Startup Methodology in order to guide the development process, and employs D&M’s IS success model to evaluate both specific features and the overall system being developed. Our goal was to prioritize development efforts based on continuous feedback from users, reducing wasted effort and increasing the value delivered to customers.

We underline two aspects: The first is that we must establish that we are developing more than a software product; we are developing an information system and, as such, we must consider the relations between software and users; system development must be approached in an holistic way and include the necessary activities beyond software development. The second is that we must look at information systems development as closer to a discovery and design and activity rather than an plan-execution
activity.

To summarize what we believe to be our main contributions:

- The identification and description of the inadequacy of traditional project management success criteria for many IS development projects;
- The identification and description of the need to switch from a plan and execute to a discover and design mindset when addressing many of today’s IS projects;
- A new approach to evaluate the success of IS projects based on the value they create for users;
- An IS development methodology that augments Scrum with a mechanism for measuring the value of specific features and uses this data to inform and guide subsequent development.

**A bridge between the Scrum methodology for software development and the Lean Startup methodology for product and business development** Our proposal can be seen as a bridge between Scrum and the Lean Startup methodologies. On one hand, it goes beyond Scrum in that it explicitly promotes the delivery of product increments to the end customer and not only to the product owner, in order to gather customer feedback and use it to guide development. On the other hand, it does not go as far as the Lean Startup goes in terms of delaying the implementation of a final solution for the sake of maximizing learning from customers. It can thus be seen as a compromise between these two approaches that can be applied in most of the contexts where Scrum is already used, with the goal of reducing wasted development effort and increasing the likelihood of developing a system that more accurately addresses customer needs.

The evaluation of this proposal, presented in the previous chapter, was overall positive. In any case, there are certainly things that could be improved, limitations and future work that could be pursued. We explore these aspects in Sections 9.2 and 9.3.

### 9.2 Limitations

Our proposal can be seen as an extension to the Scrum methodology. As such, it shares many of Scrum’s (and Agile methodologies’ in general) limitations.

Turk et al. [37] describe some of Agile’s major assumptions and limitations, of which the most relevant for our proposal are listed below:

- The assumption that customers and developers are co-located and available for frequent face-to-face communication;
- The assumption that software requirements evolve as software is being developed.
• The assumption that developers have the experience needed to define and adapt their processes appropriately;

• The assumption that rigorous evaluation of software artifacts (products and processes) can be restricted to frequent informal reviews and code testing;

• The assumption that software can be developed in increments;

• The assumption that there is no need to design for change because any change can be effectively handled by refactoring the code.

These assumptions give rise to some limitations. Most of these limitations, however, do not imply that Agile methodologies cannot be useful in such situations where assumptions do not hold true. Instead they represent warning signs: special attention must be given to these situations; some practices may need to be tweaked or there may even be other methodologies better suited for the problem. The limitations that most apply to our proposal are listed below:

• Limited support for distributed development environments. Sutherland et al. [38] reinforce the idea that adaptations to the original Scrum, such as "Scrum Of Scrums" or "Integrated Scrum" can lead to positive results in large and complex projects. Being based on the original Scrum, however, our solution assumes physical co-location and recognizes the limitation of not dealing with distributed development environments;

• Limited support for development involving large teams - many scrum practices, such as the different scrum meetings, assume a certain team size, beyond which communication, coordination and control mechanisms may become ineffective;

• Limited support for developing life-critical software - life-critical software demands quality standards that cannot be guaranteed in the solution we propose as it is;

• Limited support for developing large, complex, software - the assumption that code refactoring removes the need to design for change may not hold for large complex systems.

Lastly, and unrelated with Scrum practices, our proposal assumes that:

• The product can be launched to at least a representative subset of its end customers, early in the development process. This assumption may not hold true if, for example, the project’s sponsor does not agree with it;

• It is possible and feasible to gather the relevant usage metrics for the project. Privacy concerns or the effort required to build the measuring mechanisms may interfere with the capacity to gather user feedback which is essential for our proposal to work.

Contrary to what happened in the case of the limitations presented in the beginning of this section, if
these two last assumptions do not hold true, then our proposal can no longer be applied.

9.3 Future Work

In terms of future work, there are many interesting avenues that could help us better understand both the cases for which this methodology is most suited and the ways in which it could be improved. Many of these avenues are related to information systems development in general.

The most immediate research direction to explore has to do with scaling this methodology, which is a similar challenge to that faced in Scrum. Using this methodology with more complex projects would probably benefit more from using MVPs in their original sense, or what we called here "mock features". Defining an MVP is not trivial. There is also the problem of defining specific goals for each feature which is something we do not tackle in this proposal - what is the impact of defining minimum value (or system Use) thresholds for dropping the development of a specific feature? How could we do it? It would be interesting to explore the best approaches to these questions surrounding MVPs because this is a very important concept for the kind of projects we are dealing with here. In projects with very uncertain requirements and heavy focus on solution-discovery, deciding what to build, when to shift directions and how to evaluate the value of our hypothesis, are critical activities.

Another interesting topic to explore is how to spread the information collected during development to other people inside the organization but outside the development team. Information about the value of specific features or other trends in user behavior can be extremely useful for multiple roles inside the organization. How to best share this information either for short term action by other departments or for organizational learning, are interesting topics.

As mentioned in the beginning of this section, most of these aspects concern information systems development in general, which was the topic explored in this thesis and one that is expected to become ever more relevant in the future.
Bibliography


32. Getting Started with Rails — Ruby on Rails Guides.

33. Ruby on Rails Tutorial.


Appendix A

SKAN Platform screenshots - the system’s main features

Figure A.1: Homepage for the SKAN Platform website. The homepage is composed of 3 fixed sections (Recent, Highlights, Events) and 3 custom sections for user-specific keywords. The menu on left allows fast access to the main sections, to users, to info about SKAN or the current network one is exploring (in this case it's the global SKAN network), and to each specific network (e.g. Angola).
Figure A.2: Login page. Users can login with their Facebook or Google accounts. They can also create a new SKAN Platform user account. Users without an account have restricted access to some content.

Figure A.3: Example of a custom section in the homepage. Custom sections are built from keywords extracted from a user’s most frequent searches. These sections aggregate content related to the respective keyword.
Figure A.4: Events, list view. The fixed section presented in the homepage - Recent, Highlights and Events - have their own separate pages. Here we see the page for upcoming events and also a menu with the extra option of a Calendar view.

Figure A.5: Events, calendar view.
Figure A.6: Example of a user profile.

Figure A.7: User list and user search.
Figure A.8: Example of a post. A post can contain text, images, video and a pdf annex. Posts of the type 'event' also have date and location. If hyperlinks are included in the post's text, a short summary and image are fetched for the first hyperlink detected. This example shows a post with such an hyperlink preview, in addition to a user-uploaded image and text.

Figure A.9: Example of comments on a post. Users can comment on posts.
Figure A.10: Users can search for posts.

Figure A.11: Creating a new post.
Figure A.12: Message inbox.

Figure A.13: Notifications dashboard. Users are notified of new messages, new comments on their posts and new upvotes or downvotes on posts and comments. In addition to these notifications, users can also be notified via email, according to preferences they can change in their user settings.
Figure A.14: Administrator settings for a given SKAN network. Administrators can edit network-specific settings such as network name, logo and simplified HTML content.

Figure A.15: Cookie policy. When visiting the website for the first time, new users are informed about the usage of cookies and the website’s cookie policy.
Figure A.16: Administration interface. SKAN administrators have access to a special interface where they can perform multiple administrative tasks such as delete bad posts or fake users.